175 PTAS



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen VII-Fascículo 80

Director:	José Mas Godayol
Director editorial:	Gerardo Romero
Jefe de redacción:	Pablo Parra
Coordinación editorial:	Jaime Mardones
	Francisco Martín
Asesor técnico:	Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, 08008 Barcelona
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-067-2 (tomo 7)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 248507
Impreso en España-Printed in Spain-Julio 1985

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 19 425 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 429 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S.A. (Paseo de Gracia, 88, 5.°, 08008 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.





Datos didácticos

La utilización de bases de datos les permite a los niños desarrollar una notable destreza en el tratamiento de información

La obtención de información de una "biblioteca electrónica" reporta muchos beneficios educativos: el acceso es inmediato, y para recuperar datos no es necesario salir del aula, ni siquiera levantarse del asiento. Se evita la posibilidad de que alguien "se lleve el libro a casa". Además, los datos informatizados se pueden actualizar fácilmente, mientras que la información retenida en los libros de texto es estática y sólo se puede modificar mediante la reimpresión. Los costos actuales de la producción de libros representan un grave problema para la provisión de material fuente, y con frecuencia las escuelas se ven obligadas a proporcionar publicaciones anticuadas. Y, lo que es aún más significativo, el elevado coste de la palabra impresa puede impedir, asimismo, la difusión de nuevos métodos educativos. En una clase dotada de ordenadores, para modificar los programas de estudio sólo es necesario cambiar los discos maestros: el reaprovisionamiento completo de la biblioteca se vuelve innecesario.

Los beneficios educativos se hacen especialmente evidentes mediante la combinación del microordenador y las telecomunicaciones para servicios de base de datos en línea. Tener, por ejemplo, la Enciclopedia Larousse en disco sería una aspiración sumamente útil. Si la información que usted deseara se pudiera hallar buscando en los índices comunes, por orden alfabético, entonces sería perfectamente adecuado tomar un ejemplar del libro de la estantería. Pero si deseara hallar todos los marsupiales *de madriguera*, por ejemplo, o confeccionar una lista de todos los países donde el porcentaje del PNB invertido en educación esté por encima de un cierto nivel, entonces compilar estos datos a mano a partir del texto sería una tarea agotadora.

La Enciclopedia Larousse, sin embargo, ocupa

Un mundo para ellos
El temor de que el uso de ordenadores en la escuela confinara a los niños al estrecho mundo de la VDU parece totalmente infundado: la práctica ha demostrado que el software para bases de datos, en particular, puede inducir al niño a realizar todo tipo de actividades ajenas al teclado. La necesidad de reunir información para entrar en la base de datos se puede abordar de muchas formas diferentes, incluyendo la de "caminar por la naturaleza", en la cual los estudiantes son estimulados para observar, registrar y clasificar elementos del medio ambiente







Steve Cross

muchos volúmenes y el espacio de almacenamiento y las amplias facilidades de búsqueda necesarias requerirían una potencia de proceso extremadamente grande; y es aquí donde cobran importancia los servicios en línea. El usuario entra en ellos con un modem estándar de 300 o 1 200 baudios y la línea telefónica, para acceder a una cantidad abrumadora de información. La base de datos Inspec, por ejemplo, compilada por el Institute of Electrical Engineers, cubre toda la gama de ingeniería eléctrica y electrónica, con más de dos millones de entradas. La Royal Society of Chemists produce Chemical Engineering Abstracts, donde se listan trabajos de investigación, artículos y otros materiales de referencia. Si bien para mantenerse al día de la investigación vale la pena disponer de las versiones impresas, es más fácil buscar las referencias específicas en línea.

Dialog, en Estados Unidos, es el mayor servicio de base de datos en línea de todo el mundo y uno de los más utilizados en el campo de la educación. Contiene 200 bases que cubren más de 100 publicaciones y 100 millones de elementos de información: extractos de todo, desde historia a zoología. Cuando usted busca en el Dialog, se le ofrecen referencias no sólo del autor y el título de cualquier informe o artículo relevantes, sino que también se le ofrece un extracto con una descripción detallada y, con frecuencia, conteniendo toda la información que necesite. Por supuesto, se puede obtener una salida impresa o bien pedir que la información impresa se le envíe, por lo general a un costo moderado. La cantidad de bases de datos en línea disponibles se está respaldando con una creciente cantidad de software para búsqueda en bases de datos, para ayudar a los usuarios a extraer y clasificar la información.

En su *Schools link*, el Prestel ofrece un enfoque diferente. A través de este servicio se les ofrece a las escuelas información sobre todos los aspectos de la informática y se crea un foro de ideas y contactos. Está dirigido a los maestros en lugar de a los alumnos, y contiene comentarios de software, robots educativos, libros y sugerencias para emprender proyectos; asimismo, facilita programas educativos para cargar en la máquina del maestro o de la escuela.

No obstante, además de la "biblioteca electrónica", las bases de datos están adquiriendo creciente popularidad en la clase, donde juegan un papel muy activo. Ya existen varios programas que le ofrecen al niño en edad escolar la experiencia de manipular información y estructurarla de acuerdo a sus propias necesidades individuales, proporcionándole al niño la oportunidad de reunir, entrar y manipular sus propios datos.

Una base de datos llamada "Factfile"

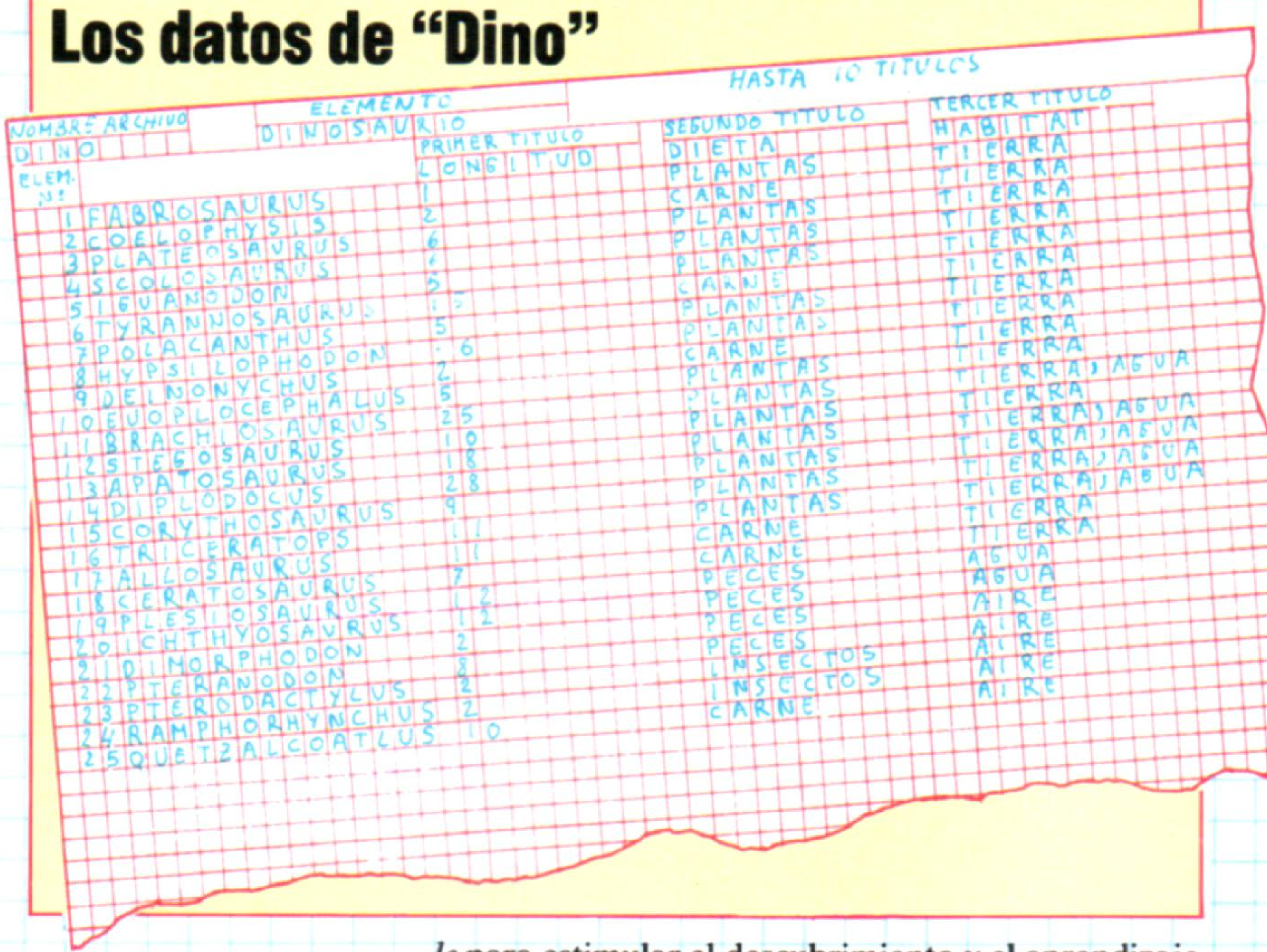
Las posibilidades del uso de las bases de datos fueron el tema de un congreso educativo que tuvo lugar en 1981 en la Universidad de Cambridge, en Gran Bretaña, y el debate inspiró la creación de un programa llamado *Factfile* (Archivo de hechos), que permite a los niños pequeños construir archivos de datos sobre cualquier tema que elijan.

El papel activo de una base de datos como *Factfile* para estimular el descubrimiento y el aprendizaje se hizo evidente enseguida. Supongamos que los niños desean crear una base de datos que contenga información sobre sus maestros. Primero tendrán que pensar en las categorías por las que van a organizar la información. Esto dependerá de lo que ya sepan y de lo que puedan descubrir por sí mismos. Podrían decidir, por ejemplo, incluir edad, altura, asignatura, sexo y "tipo de maestro". Habiendo determinado la estructura de la base de datos, deben luego reunir información para las categorías específicas. En este punto, el ejercicio se convierte en una operación educativa variada: como mínimo, los niños tendrán que reunir información física. Si se hubieran de realizar mediciones, deberán determinar las unidades a utilizar y los datos a incluir. El proceso adquiere mayor complejidad puesto que el niño comienza a comprender el significado de las categorías que ellos mismos han generado. El campo "tipo de maestro" exige un buen nivel de análisis: significa "¿Qué asignatura enseña?" o "¿Es el maestro agradable o antipático?".

Cuando hayan terminado de compilar la base de datos, otros niños pueden utilizarla, correlacionando información de más de una categoría. Quizá deseen listar todas las maestras que se han clasificado en la categoría "agradable" o todos los maestros de matemáticas. El proceso de aprendizaje da otro paso más hacia adelante cuando el niño empieza a analizar los resultados. *Factfile* permite que los niños guarden su base de datos en disco o cassette y añadan información posteriormente, permitiéndoles retornar a los archivos siempre que así sea necesario.

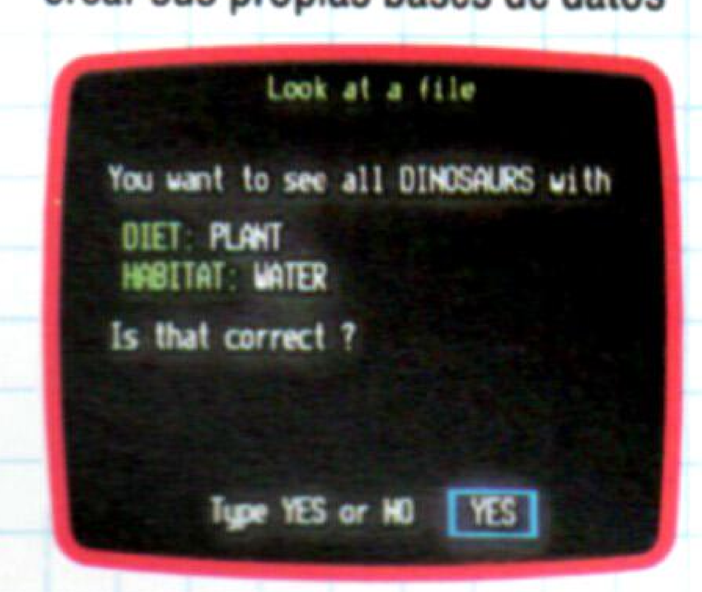
Una crítica que se le suele hacer al uso del ordenador por parte de los niños es que el medio los estimula a permanecer sentados delante de la VDU desarrollando actividades que no guardan relación con el mundo en el que viven. Pero *Factfile*, aparte de facilitarle al alumno una experiencia directa con bases de datos, favorece numerosas actividades ajenas al teclado. Si se crea un archivo con detalles de la fauna local, los niños tendrán que registrar cuidadosamente los detalles de cada planta y después de-

Los datos de "Dino"

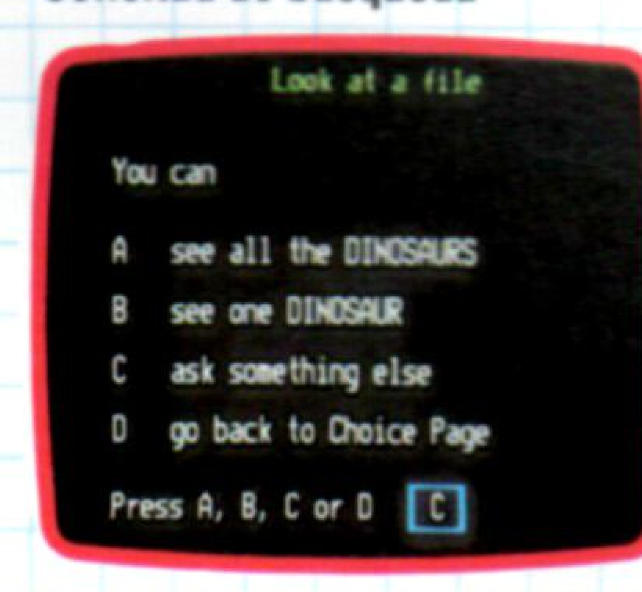


ELEM. Nº / DINOSAURIO	PRIMER TITULO: LONGITUD	SEGUNDO TITULO: DIETA	TERCER TITULO: HABITAT
1 FABROSAURUS	1	PLANTAS	TIERRA
2 COELOPHYSIS	2	CARNE	TIERRA
3 PLATEOSAURUS	6	PLANTAS	TIERRA
4 SCOLOSAURUS	6	PLANTAS	TIERRA
5 IGUANODON	5	PLANTAS	TIERRA
6 TYRANNOSAURUS	5	CARNE	TIERRA
7 POLACANTHUS	5	PLANTAS	TIERRA
8 HYPSILOPHODON	6	PLANTAS	TIERRA
9 DEINONYCHUS	2	CARNE	TIERRA
10 EUOPLOCEPHALUS	5	PLANTAS	TIERRA
11 BRACHIOSAURUS	25	PLANTAS	TIERRA, AGUA
12 STEGOSAURUS	10	PLANTAS	TIERRA
13 APATOSAURUS	18	PLANTAS	TIERRA, AGUA
14 DIPLODOCUS	28	PLANTAS	TIERRA, AGUA
15 CORYTHOSAURUS	9	PLANTAS	TIERRA, AGUA
16 TRICERATOPS	11	PLANTAS	TIERRA, AGUA
17 ALLOSAURUS	11	CARNE	TIERRA
18 CERATOSAURUS	7	CARNE	TIERRA
19 PLESIOSAURUS	12	PECES	AGUA
20 ICHTHYOSAURUS	12	PECES	AGUA
21 DIMORPHODON	2	PECES	AIRE
22 PTERANODON	8	PECES	AIRE
23 PTERODACTYLUS	2	INSECTOS	AIRE
24 RAMPHORHYNCHUS	2	INSECTOS	AIRE
25 QUETZALCOATLUS	10	CARNE	AIRE

Hace un millón de años...
Factfile es un sistema completo de base de datos para los estudiantes más jóvenes, que representa una excelente introducción a la administración de bases de datos. Con el programa se entrega un archivo muestra, llamado "Dino", que contiene información sobre los dinosaurios. Se estimula a los niños para que se familiaricen con el "Dino" y cataloguen el contenido del archivo (que vemos arriba) antes de pasar a crear sus propias bases de datos



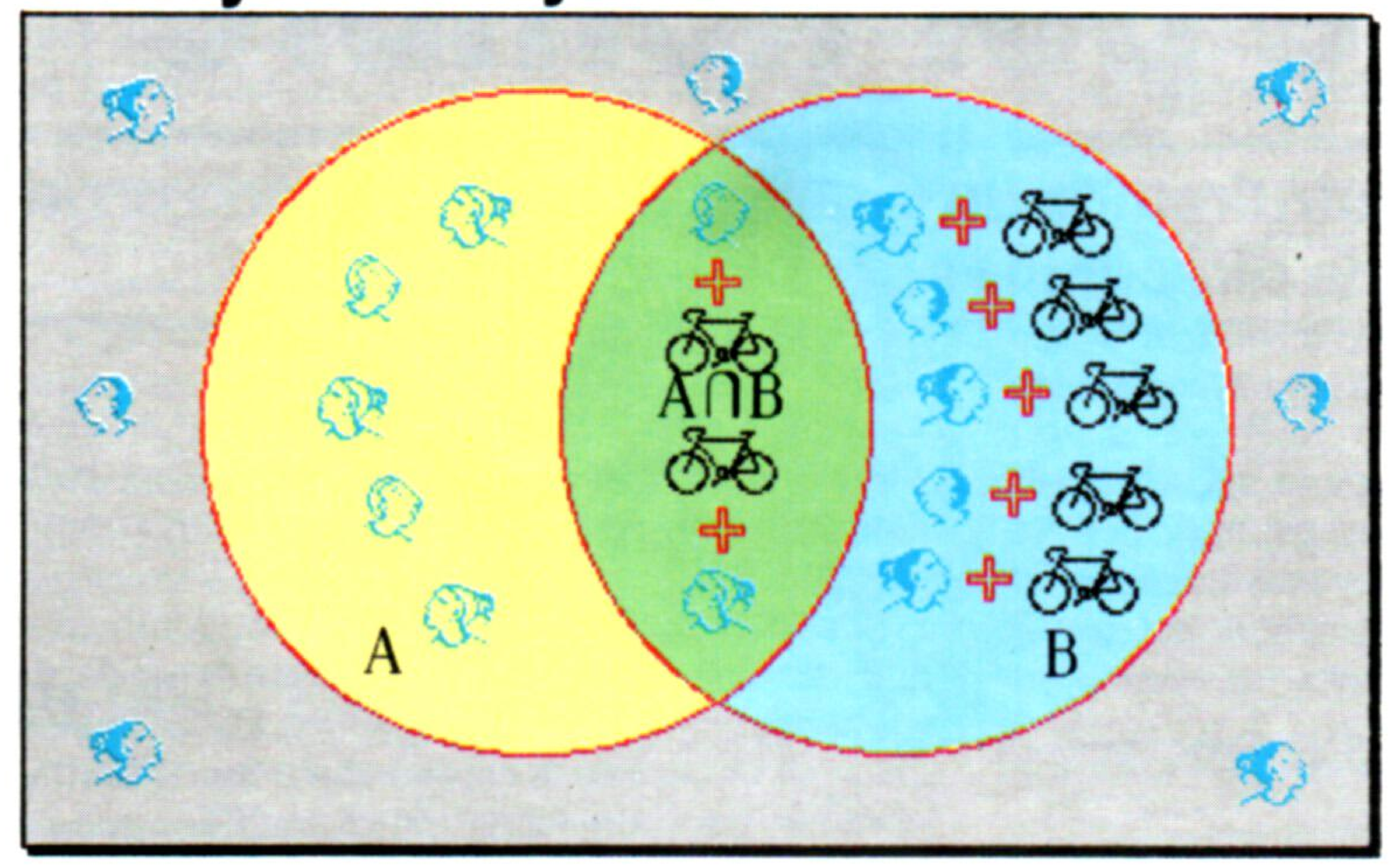
Solicitud de búsqueda

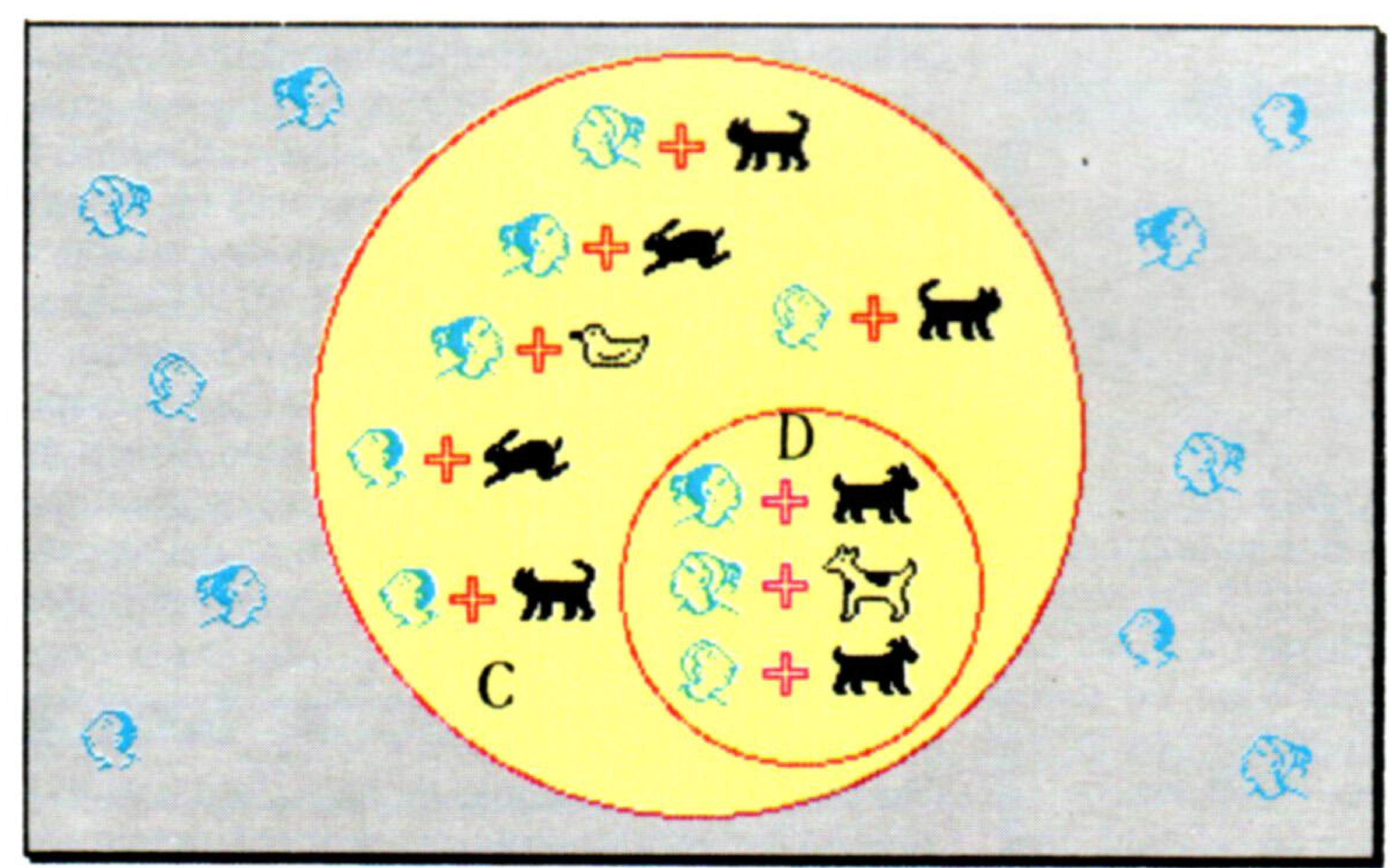


Menú del "Dino"



Trabajo de conjuntos





Ian McKinnell

sarrollar su propio sistema de clasificación al objeto de construir la base de datos. En una escuela de Londres se construyó una base de datos como parte de un programa de estudios sobre el medio ambiente, utilizando como entrada los resultados de una encuesta efectuada entre personas de aquella zona de la capital británica.

Your facts (Tus hechos), diseñado para niños más pequeños, es otro interesante paquete. Al pequeño se le pregunta su nombre, si es niño o niña, si tiene algún animalito, un reloj, una bicicleta y un hermano o hermana. Después de digitarse la información concerniente a un alumno, el programa pregunta si a alguien más le gustaría digitar su propia información. Hay lugar para datos de unos 40 niños, por lo cual se puede incluir a toda la clase. Una vez digitada toda la información, y tras haberse digitado "No" en respuesta a "¿ALGUIEN MAS QUIERE HACER UNA ENTRADA?", los niños retornan al menú, donde pueden ver todos los registros, descubrir qué niños caen en una categoría determinada (p. ej., quiénes tienen un reloj o quiénes tienen una bicicleta), o bien pueden participar en un juego. Éste consiste en que el ordenador adivina el nombre de un niño tras formular varias preguntas como "¿ERES UN NIÑO?", "¿TIENES UN RELOJ?", etc. Luego pregunta, por ejemplo, "¿TE LLAMAS TIMOTHY?".

Your facts constituye para el niño pequeño una introducción ideal al concepto de lo que es una base de datos. Fomenta la lectura y la escritura, establece con el niño una relación personal y es un ejercicio entretenido.

Información clasificada
La construcción y la interrogación de bases de datos guarda cierta relación con otras tareas que realizan los niños en la escuela. Los programas de estudio de matemática moderna introducen la noción de conjunto a una edad muy temprana. La teoría de conjuntos guarda una estrecha relación con la clasificación de datos y las relaciones entre tales clasificaciones. Preparar una base de datos es, en muchos sentidos, una labor directamente análoga, categorizando las propiedades de una entrada en campos y registros.
Los conceptos de intersección y subconjunto, en especial, poseen un verdadero significado en la interrogación de bases de datos, correspondiendo a la creación de listas cortas. La intersección es la "superposición" de dos conjuntos: por ejemplo, la intersección de un conjunto de niños rubios con el conjunto de niños que tienen bicicleta es el conjunto de niños rubios que poseen bicicleta. Un caso especial de intersección es aquel en el que un conjunto se superpone completamente con otro, formando un subconjunto, como en el segundo ejemplo

El programa "One world"

Una empresa de software australiana, Active Learning Systems, ha producido un amplio paquete de base de datos de información relativa a casi todos los países del mundo. El nombre del programa es *One world* (Un solo mundo) e incluye observaciones y hojas de trabajo docentes para el individuo y para la clase. Contiene más de 30 elementos de información para cada país, incluyendo datos sobre el sistema político, importaciones, exportaciones, idiomas, países limítrofes, religión, tasa de alfabetismo, antecedentes históricos y tratados. Se da el porcentaje de población rural y urbana, así como porcentajes de la fuerza de trabajo en la producción, manufactura y principales industrias de servicios. También se incluyen las proporciones de zonas desérticas, bosques y tierra cultivada. El atractivo del programa, al igual que el de la mayoría de las bases de datos, reside en sus facilidades para buscar y analizar los datos.

El menú principal ofrece al usuario una serie de opciones:

1. VISUALIZAR CUALQUIER PAIS
2. BUSCAR UTILIZANDO INFORMACION UNICA
3. ANALIZAR UTILIZANDO DIVERSOS CRITERIOS
4. REFERENCIA DE AYUDA
5. SALIR

Si un niño conociera la existencia del señor Shamir, un líder del Oriente Medio, y quisiera averiguar su país, seleccionaría la segunda opción. El menú BUSCAR DATO permitiría que el niño digitara su nombre en 3.JEFE DE GOBIERNO y 4.JEFE DE ESTADO. El programa le informaría entonces que YITZHAK SHAMIR ocupaba el cargo de jefe de estado de ISRAEL.

La opción ANALIZAR es la más brillante del programa. Permite que el niño analice información, utilizando hasta tres opciones, entre 20 criterios. Por ejemplo, podría averiguar en qué países europeos la mayoría de la población vive en el campo. Ello sólo emplearía dos opciones: región y porcentaje de población. Si el niño se entera de que los mineros de bauxita han ido a la huelga en Surinam, rápidamente puede descubrir la situación del país, que la bauxita es una de las principales exportaciones y que la huelga puede tener un efecto devastador en la economía.

One world enseña las técnicas de la investigación e interpretación de datos, y estimula a los niños a sacar conclusiones a partir de los análisis que hagan de la información suministrada. Constituye un gran apoyo para las lecciones de historia, geografía y ciencias sociales.

Una base de datos puede indicarle al alumno la mayoría de lo que hay para aprender y lo estimula a buscar las respuestas. El niño puede observar cómo manipula la información el ordenador y comprender la importancia de formular las preguntas correctas y organizar adecuadamente las entradas. Quienes, tras haber adquirido experiencia en el uso de una base de datos, accedan a la educación superior y finalmente al mercado de trabajo, con toda seguridad poseerán una destreza verdaderamente importante.

Nombre por nombre

En esta ocasión construiremos un sistema sencillo basado en la idea de una agenda de direcciones

Una de las principales ventajas de la agenda de direcciones tradicional es que, a excepción de la organización por orden alfabético, hay muy poca estructuración. Es flexible en el sentido de que uno puede tener entradas tales como:

PEDRO FERNÁNDEZ, Pedreña 809, Madrid (91-3397881)
— Paloma, su novia — despacho: 91-3388771 ext 160
— llamar después de las 4 — en Ene. se muda a un piso nuevo.

Seguida de:

FIDEL — 396-1949

Seguida de:

FINA — ver Bermúdez

Utilizando una agenda de direcciones, uno está en libertad de incluir a las personas por su nombre de pila, su apellido o incluso mediante "direccionamiento indirecto", con "apuntadores" a otras entradas. Sin embargo, cuando una base de datos como ésta se transfiere a un ordenador, por lo general se requiere un enfoque más sistemático.

Dado que, en comparación con la original, una agenda de direcciones informatizada posee algunas desventajas, vamos a diseñar el formato para una que sea lo suficientemente flexible como para hacer frente a la mayoría de las circunstancias. Consideraremos cada campo uno por uno, empezando por el del nombre.

El nombre tiene dos partes: la mitad genérica, denominada apellido, y la mitad específica, denominada nombre de pila. En la mayoría de los países, el o los nombres de pila van seguidos por el apellido, mientras que en China, Japón, Hungría y otros países, el apellido va antes del nombre de pila. En Japón, hay sólo un nombre de pila, mientras que en la comunidad china de Inglaterra se acostumbra a tener un nombre de pila inglés además del nombre de pila chino. De este modo, Li (apellido) Yu Chow (nombres de pila) para la mayoría de sus amigos ingleses será Paul.

Con esto ya tenemos muchísimas probabilidades de que se produzcan confusiones, y eso que sólo estamos teniendo en cuenta el campo del nombre. Es evidente que hemos de imponer alguna disciplina y decidir qué irá primero, si los nombres de pila o los apellidos, y qué longitud podrá concederse a un nombre. Los nombres pueden ser tan cortos como Ng o tan largos como Cholmondley-Smythe, y posiblemente incluso más cortos o más largos, de modo que siempre debemos dar cabida a los casos más extremos. Si el administrador de la base de datos (DBM) utiliza campos de longitud fija, tendremos que elegir una longitud de campo adecuada para el nombre más largo con el que podamos encontrarnos. (Más adelante, en esta serie, hablaremos de las desventajas que suponen los campos de longitud fija.)

Si hemos de imponer un formato, como en realidad debemos hacer al diseñar una base de datos, probablemente lo más simple sea utilizar el campo del apellido como el campo clave fundamental, de modo que comenzamos con el apellido, que irá seguido por uno o más nombres de pila. La dirección, asimismo, plantea problemas. Indudablemente, usted habrá visto en revistas norteamericanas formularios para recortar que piden su Ciudad, Estado y Código Postal. Las direcciones en España no se ajustan a ese formato. Otra complicación surge al considerar que incluso el formato "Nombre, Número de la Casa, Calle, Ciudad, Provincia, País" tampoco es apropiado. En Japón, por ejemplo, la ciudad ocupa el primer lugar en una dirección, seguida por el distrito de la ciudad, seguida por el número del solar del edificio dentro del distrito, viniendo en último lugar el nombre del destinatario.

Una especificación básica

En este sencillo ejemplo de base de datos, al igual que en las más complejas, probablemente no sea posible hacer frente a todas las combinaciones que se puedan concebir, de modo que habremos de llegar a un compromiso. Suponiendo que no es probable que tenga que comunicarse alguna vez con alguien que tenga 16 nombres de pila, veamos cómo funcionaría el siguiente esqueleto básico:

- Campo del Apellido: hasta 40 caracteres
- Campo del Nombre de Pila: hasta 60 caracteres
- 1.ª Línea Campo de Dir.: hasta 80 caracteres
- 2.ª Línea Campo de Dir.: hasta 80 caracteres
- 3.ª Línea Campo de Dir.: hasta 80 caracteres
- 4.ª Línea Campo de Dir.: hasta 80 caracteres
- 5.ª Línea Campo de Dir.: hasta 80 caracteres
- Campo del Teléfono: hasta 20 caracteres
- Campo de Notas: hasta 80 caracteres

Estas especificaciones básicas cubrirán la mayoría de las eventualidades, aunque aún podrían plantearse problemas. Una dificultad potencial es la limitada longitud del Campo de Notas. Otro problema es que País no está especificado como un campo separado. Según la longitud de la dirección, el país puede hallarse en el tercero, cuarto o quinto campo de la dirección, o en ninguno en absoluto. Ello normalmente no representaría ningún problema hasta que deseáramos buscar en nuestra base de datos utilizando País como clave. Si su aplicación de base de datos ha de tratar con muchos corresponsales extranjeros, será mejor diseñar la base con un campo de País separado. Decisiones de esta clase siempre serán necesarias en la etapa de diseño.

El DBM que ejecute en su ordenador determina-

rá la facilidad con la que pueda implementar una base de datos en su sistema. Uno de los DBM más simples es el *Card box* de Caxton. El programa posee facilidades limitadas para extraer y manipular datos, pero si sus exigencias son normales, le dará los resultados que necesite con un mínimo trabajo. *Card box* no ofrece un lenguaje de programación sofisticado para manipular campos ni registros. Sí permite, sin embargo, que el usuario extraiga registros específicos entrando desde el teclado instrucciones simples. Utilizando el *Card box* para crear la base de datos esbozada más arriba, usted sólo habrá de cargar el programa y seguir una sencilla secuencia de entradas.

Uso del "Card box"

Lo primero que debe hacer es decidir el formato del registro. Este formato determina qué información se almacenará, cómo se indexará (es decir, qué campos se designarán como campos "clave") y cómo aparecerán finalmente los registros en la pantalla del ordenador o en las salidas impresas. Si al archivo de la base de datos lo llamamos AGENDADIR, *Card box* creará un archivo de formato denominado AGENDADIR.FMT. Éste se puede editar si fuera necesario modificar la forma en que se visualiza la información. También se pueden crear archivos de formato alternativos para visualizar la información de la base de datos de formas diferentes.

Al igual que la mayoría de los DBM, *Card box* permite entrar texto *permanente* en cada uno de los campos. En el caso de una agenda de direcciones, es bastante evidente cuál es el significado de cada campo: Juan Pérez es a todas luces un nombre y no parte de una dirección; de modo similar, 339-44-20 es, obviamente, un número de teléfono. En otras bases de datos, quizá deba recordar cuál es el significado de cada campo. Es allí donde entra en juego el texto permanente. Compare estos dos registros:

06116
3995
86
34.75
Paquete 100 folios

y

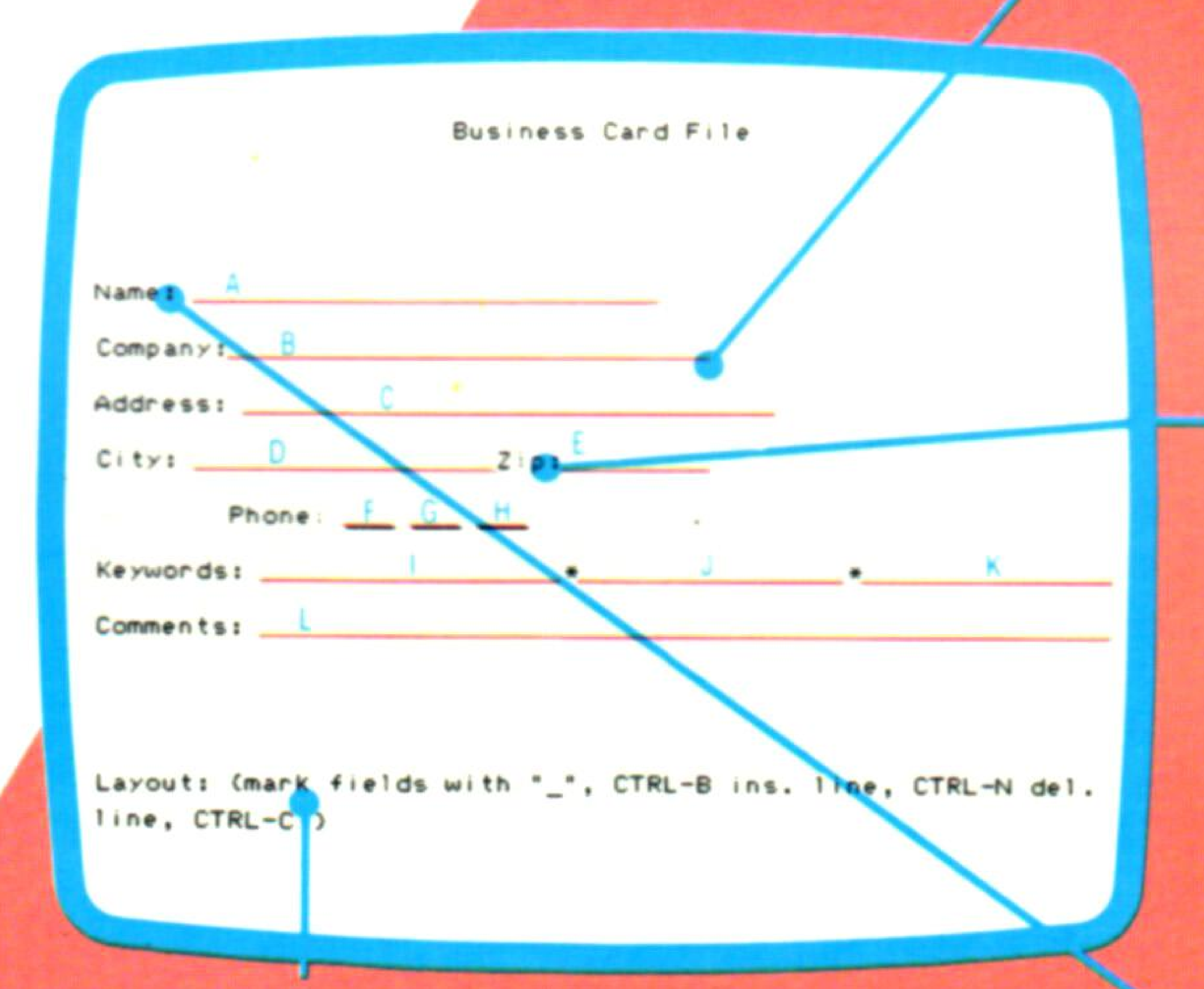
NUMERO DE COMPONENTE DEL FABRICANTE	06116
NUMERO DE COMPONENTE NUESTRO	3995
CANTIDAD EN STOCK	86
PRECIO	34.75
DESCRIPCION	Paquete 100 folios

La información es idéntica en ambos casos, pero en el segundo ejemplo, con la aparición de un texto permanente en cada registro, es mucho menos probable que se produzcan errores.

Card box permite dar a cada campo uno de cuatro atributos posibles: NONE (ninguno), MAN(ual), AUTO o ALL (todos). En una base de datos de stock, es improbable que alguna vez sea necesario usar PRECIO como campo clave: no es probable que usted pregunte: "¿Tiene algún componente que cueste menos de 600 ptas. y más de 400 ptas?" No obstante, es bastante posible que necesitemos saber cuántos artículos #06116 hay todavía en stock, de modo que será necesario que NUMERO DE COMPONENTE DEL FABRICANTE sea un campo clave.

Retomando nuestro ejemplo de la agenda de direcciones, es seguro que desearemos buscar en la base de datos en función de Nombre y, probablemente, también de Nombre de Pila. Ambos habrán de ser campos clave. Si estuviéramos llevando una empresa, podríamos necesitar, asimismo, buscar registros por Ciudad y posiblemente incluso por Código de Distrito Telefónico. Es importante recordar que no se puede crear una base de datos eficaz a menos que uno haya diseñado perfectamente cómo habrá de emplearse la base. A diferencia de los ficheros de tarjetas tradicionales, las bases de datos exigen que uno prevea de antemano cómo se la utilizará en el futuro.

Elementos de diseño
Megafinder permite al usuario diseñar sus propios formularios. Éste se obtuvo a partir del mismo formulario que se proporciona ya hecho con el programa, que es el que utilizamos en nuestra otra visualización



LONGITUD DE CAMPO
Se indica mediante caracteres de subrayado (__). La longitud total de todos los campos de un formulario determina la cantidad de formularios que se pueden almacenar en un archivo. Por consiguiente, resulta práctico mantener los campos tan cortos como sea posible, teniendo presente la naturaleza de los datos a entrar

ETIQUETAS DE CAMPO
Se las entra al diseñar el formulario; en este programa se incluyen sólo para comodidad del usuario. El programa las emplea cuando visualiza información en la pantalla, pero no tiene en cuenta las etiquetas de los campos cuando clasifica datos, tarea que realiza utilizando IDENTIFICADORES DE CAMPO

IDENTIFICADORES DE CAMPO
Los identificadores de campo, que no se deben confundir con las ETIQUETAS DE CAMPO, determinan el orden por el cual se accederá a los campos cuando se entre información. Asimismo, los identificadores permiten clasificar los campos por separado o en grupos. Utilizando la opción INDEX (índice) de *Megafinder*, la asignación del campo D al Indice 1 le permitiría al usuario clasificar rápidamente los diferentes registros de acuerdo a la disposición alfabética de ciudades

LÍNEA DE INSTRUCCIONES
Muestra algunas de las instrucciones que se pueden utilizar al editar un diseño. Se pueden insertar y suprimir líneas y caracteres, y el diseño terminado se puede guardar en disco pulsando CRTL-C

```
     Company: Home_Computer_Advanced_Course__________
     Address: 14_Rathbone_Place______________________
     Town:LONDON___________________Code:W1P_1DE____
     Telephone:01_631_1433_______
     Contact:STEPHEN_COOKE ________\STEVE_MALONE_______

Keywords: PUB____ * WEEKLY_ * 1.00___

Product received:Subscription____________________
Actions: 25/5/85_renew_sub_______________________
         ________________________________________

Find:  I(ndex 1 = ACD  M(atch = OFF (arrows) J(ump C(hange
D(elete P(rint Q(uit
```

Permítame que le enseñe mi tarjeta
Este vuelco de pantalla del *Megafinder*, un DBM que se ejecuta en el Apple, muestra un registro seleccionado de un archivo llamado "Tarjetas de empresas" y, debajo del mismo, una lista de instrucciones opcionales. Los datos retenidos por el programa se pueden clasificar en hasta cuatro índices diferentes. El DBM también puede buscar una pareja (MATCH) entre los datos entrados y los almacenados, saltar (JUMP) a secciones especificadas de la base de datos (p. ej., a la que retiene nombres de empresas que empiezan con M), cambiar (CHANGE) y suprimir (DELETE) información, e imprimir (PRINT) un registro o un listado

Levando anclas

Finalizados los preparativos, ya podemos zarpar y desarrollar un módulo que se encargue del progreso semanal de la travesía

Las incidencias semanales durante el viaje se pueden controlar desde un simple bucle FOR...NEXT, utilizando un contador de bucle, WK, para contar las semanas del viaje. El límite superior del bucle se establece mediante el valor de la variable JL, cuyo valor al comienzo del programa se establece en ocho. Desde el bucle podemos llamar a subrutinas para tratar cada una de las tareas que se produzcan sobre una base semanal.

La dieta es un factor fundamental para determinar la fortaleza de un tripulante. Para mantenerse en forma, éste debe consumir los requerimientos semanales de fruta, verduras, carne y agua. Al comienzo del viaje, la fortaleza de cada miembro de la tripulación se establece en 100 en la matriz TS(,). El programa hace un cálculo semanal para comprobar si hay suficientes existencias de cada alimento para toda la travesía. De descender las existencias de algún tipo determinado, se le dice al jugador que la tripulación debe empezar a consumir media ración de ese alimento. Las medias raciones sólo se suministrarán durante una semana y reducirán en cinco unidades la fortaleza de la tripulación.

De acabarse algún tipo de alimento, la fortaleza de cada tripulante se reduce en 10 unidades por cada semana que pasen sin él. Por ejemplo, si la tripulación se quedara sin fruta y estuviera a media ración de agua, su tasa de fortaleza semanal se reduciría en 15. Si el agua se terminara por completo, la tasa de fortaleza se reduciría entonces en 20 por semana. Si, tras estar a media ración durante una semana, quedaran suficientes alimentos o agua para volver a las raciones completas durante el resto del viaje, se repartirán automáticamente al comienzo de la semana siguiente.

El estado de salud de la tripulación se da al comienzo de cada semana mediante la subrutina de la línea 4000. Hay cuatro estados de salud, determinados por la tasa de fortaleza. La clasificación más alta es *muy sano* (tasa de fortaleza 75-100), seguida de *sano* (50-75), *enfermo* (25-50) y *muy enfermo* (1-25). Cuando la tasa de fortaleza llega a 0, el tripulante muere y es enterrado en el mar, pero los salarios del fallecido se seguirán teniendo en cuenta cada semana y, cuando el barco llegue a puerto, se le abonarán al pariente más próximo. Esta subrutina también da la factura de los salarios para la próxima semana, los salarios totales del viaje hasta ese momento y el dinero que queda. El total de salarios acumulado se establece a cero al comienzo de esta sección, en la línea 800, y disminuye en función de la factura salarial semanal cada vez que se efectúa el bucle. Si la factura se vuelve mayor que el saldo de oro (comprobado por la subrutina de la línea 5000), se le advierte al jugador que se necesitará parte de los beneficios de la actividad comercial para pagar a la tripulación.

Si todo va bien el viaje durará ocho semanas, pero circunstancias imprevistas pueden aumentar este valor. La línea 801 crea una variable en serie, H$, para indicar si la tripulación se halla a medias raciones. La rutina determina la fortaleza y categoría de cada tripulante mediante la comprobación de la matriz de categoría/fortaleza, TS(,). Prepara un bucle en la línea 4060 para cada posible miembro. En la matriz hay espacio para información sobre 16 tripulantes, de modo que el bucle se establece en 16. Si la categoría de tripulación está registrada como 0, esa sección de la matriz estará vacía, de modo que la línea 4070 lleva al programa hasta la línea 4110. Si en la matriz hay registrado un tripulante, la línea 4075 imprime la descripción.

Cuando la salud de un tripulante llega a 0, la muerte debe registrarse en el diario de navegación del capitán. Puesto que una tasa de fortaleza de cero indica la muerte, cada sección vacía de la matriz parecería indicar un tripulante fallecido. Para evitar este problema, cuando una tasa alcanza el valor 0 se la restablece a −999 durante una semana. Al final de la semana se vuelve a establecer en 0 y al comienzo de la semana siguiente el programa informa sobre el fallecimiento. El restablecimiento en 0 hace que el programa ignore esa tasa durante las siguientes semanas. Ello evita que se imprima al

de un factor de 10. Esta reducción la maneja realmente una pequeña subrutina en la línea 9300, que reduce la tasa de fortaleza para cada tripulante en función del valor de WF. Esta subrutina, por consiguiente, se puede utilizar para reducir la fortaleza de la tripulación en cantidades variables. Antes de llamar a la subrutina, sólo hemos de establecer WF en el valor deseado.

Si la cantidad de provisiones restantes es mayor que cero, entonces se realiza otra comprobación para ver si quedan provisiones suficientes para el resto del viaje. Esta cantidad se calcula en la línea 5180 a partir de la cantidad de tripulantes, CN, las necesidades semanales de alimentos, PN(), y el número de semanas que aún faltan para terminar la travesía. Si la cantidad de provisiones restantes es menor que las exigencias planificadas, se le da al jugador la opción de poner a la tripulación a media ración de ese tipo de alimento. El estado de cada tipo de alimento (ya sea ración completa o media ración) se indica mediante la matriz HR(), DIMensionada en la línea 802. Cuando las raciones son completas, el elemento de la matriz se establece en 1; si el alimento determinado se reduce a la mitad, entonces se establece en 0.5 el elemento correspondiente de la matriz. Si la tripulación se halla a media ración, entonces se emplea la subrutina de la línea 9300 para disminuir en cinco la tasa de fortaleza de cada miembro de la tripulación.

La comprobación final que efectúa esta subrutina antes de repartir la ración es ver si la cantidad de alimento que queda es menor que la cantidad a repartir. Si así fuera, sólo se repartiría la cantidad disponible y, para indicar que no queda ninguna provisión, el elemento correspondiente de la matriz de provisiones se establecería en −999.



empezar cada semana un registro de los tripulantes fallecidos. Las líneas 4080-4098 verifican la fortaleza de los tripulantes restantes. Sus tasas se cotejan con los cuatro estados de salud, y la línea 4099 imprimirá el estado de cada uno. El bucle fortaleza-/categoría termina en la línea 4110, que vuelve a enviar el programa a examinar la siguiente sección de la matriz.

La factura salarial semanal se calcula utilizando otro bucle entre las líneas 4120 y 4135 y se emplea una vez para cada una de las categorías de tripulantes. Los salarios para cada categoría se calculan mediante la ecuación de la línea 4130, multiplicando el contador para cada categoría por la tasa salarial, y acumulándolo en la variable WW. Creada y establecida en 0 al comienzo de cada semana mediante la línea 4119, WW representa el salario semanal, que se imprime en la línea 4145. Luego se suma al total para el viaje hasta ese momento (WT) y se imprime en las líneas 4160 y 4165. El jugador puede compararlo con el dinero que aún le queda, que se imprime en la línea 4175.

La subrutina de la línea 5100 maneja el reparto de raciones entre la tripulación. Utilizando un bucle FOR...NEXT para repetir el proceso para cada tipo de ración, se realizan varias comprobaciones del nivel de raciones restantes. La primera de ellas es una comprobación para asegurar que queden raciones. Esto se puede verificar simplemente comprobando que la cantidad de alimento, retenida en PA(), sea mayor que cero. De no ser éste el caso, la fortaleza de cada tripulante se reduciría en función

Complementos al BASIC

Spectrum:
Realice las siguientes modificaciones:

```
 847 LET I$=INKEY$:IF I$="" THEN GO TO 847
4010 CLS
4190 LET I$=INKEY$:IF I$="" THEN GO TO 4190
4205 CLS
4295 LET I$=INKEY$:IF I$="" THEN GO TO 4295
4305 CLS
4398 LET I$=INKEY$:IF I$="" THEN GO TO 4398
5010 CLS
5080 LET I$=INKEY$:IF I$="" THEN GO TO 5080
5103 CLS
5220 INPUT I$:LET I$=I$(1 TO 1)
5298 LET I$=INKEY$:IF I$="" THEN GO TO 5298
```

BBC Micro:
Realice las siguientes modificaciones:

```
4010 CLS
4190 I$=GET$
4205 CLS
4295 I$=GET$
4305 CLS
4398 I$=GET$
5010 CLS
5298 I$=GET$
```

Módulo cuatro: El viaje

Longitud de la variable del viaje

```
40 JL=8:REM DURACION DEL VIAJE
```

Bucle principal del viaje

```
800 WT=0
801 H$="N": REM INDICADOR DE MEDIA RACION
802 DIM HR(4):HR(1)=1:HR(2)=1:HR(3)=1:HR(4)=1
820 FOR WK=1 TO JL : REM BUCLE PRINCIPAL DEL VIAJE
825 GOSUB 4000: REM INFORME ESTADO TRIPULACION
830 GOSUB 4200: REM INFORME PROVISIONES
835 GOSUB 4300: REM INFORME OTRAS MERCANCIAS
840 GOSUB 9200: PRINTCHR$(147)
842 PRINT:PRINT:PRINT
843 S$="SE CALCULA QUE EL VIAJE*":GOSUB 9100
844 PRINT"DURARA AUN OTRAS";JL-WK+1;"SEMANAS"
845 GOSUB 9200
846 PRINT:S$=K$:GOSUB 9100
847 GET I$:IF I$=""THEN 847
850 GOSUB 5000:REM COMPROBAR FACTURA SALARIAL
855 GOSUB 5100:REM REPARTIR RACIONES
889 NEXT WK
```

Informe sobre el estado de la tripulación

```
4000 REM INFORME SOBRE EL ESTADO DE LA TRIPULACION
4010 PRINTCHR$(147)
4020 S$="          DIARIO DE NAVEG. DEL CAPITAN*":GOSUB 9100
4025 S$="          _______________________________*":GOSUB 9100
4030 GOSUB 9200
4035 PRINT"AL EMPEZAR LA SEMANA";WK
4040 S$="EL ESTADO DE LA TRIPULACION ES:*":GOSUB 9100
4045 GOSUB 9200:PRINT
4055 PRINT
4060 FOR T=1 TO 16
4070 IF TS(T,1)=0 THEN 4110
4075 PRINT C$(TS(T,1));"     (";
4078 IF TS(T,2)=-999 THEN S$="MUERTO !!!!!!)*":GOTO 4099
4080 IF TS(T,2)>75 THEN S$="MUY SANO)*":GOTO 4099
4085 IF TS(T,2)>50 THEN S$="SANO)*":GOTO 4099
4095 IF TS(T,2)>25 THEN S$="ENFERMO  !  )*":GOTO 4099
4098 S$="MUY ENFERMO !!)*"
4099 GOSUB 9100:GOSUB 9200
4110 NEXT T
4115 GOSUB 9200:PRINT
```

Factura salarial semanal

```
4119 WW=0
4120 FOR T=1 TO 5
4130 WW=WW+(CC(T)*WG(T))
4135 NEXT
4140 S$="FACTURA SALARIAL PARA LA SEMANA*":GOSUB 9100
4145 PRINT WW;"PIEZAS DE ORO"
4150 GOSUB 9200
4155 WT=WT+WW
4160 S$="TOTAL SALARIOS DEL VIAJE HASTA AHORA:*":GOSUB
     9100
4165 PRINT WT;"PIEZAS DE ORO"
4170 GOSUB 9200
4175 PRINT"DINERO RESTANTE=";MO;"PIEZAS DE ORO"
4180 PRINT:S$=K$:GOSUB 9100
4190 GET I$:IF I$=""THEN 4190
4199 RETURN
```

Informe sobre provisiones

```
4200 REM INFORME SOBRE PROVISIONES
4205 PRINTCHR$(147)
4206 PRINT"AL EMPEZAR LA SEMANA";WK:GOSUB 9200
4210 S$="TE QUEDAN LAS SIGUIENTES*":GOSUB 9100
4215 S$="PROVISIONES:*":GOSUB 9100
4220 PRINT:GOSUB 9200
4225 FOR T=1 TO 4
4226 IF PA(T)=0 OR PA(T)=-999 THEN 4240
4230 PRINT PA(T);U$(T);"S DE ";P$(T)
4232 X=INT(PA(T)/(CN*PN(T)))
4235 PRINT"(SUFICIENTE PARA";X;"  SEMANAS)"
4239 GOSUB 9200
4240 NEXT
4290 PRINT:S$=K$:GOSUB 9100
4295 GET I$=IF I$=""THEN 4295
4299 RETURN
```

Informe sobre otras mercancías

```
4300 REM INFORME OTRAS MERCANCIAS
4305 PRINTCHR$(147)
4306 PRINT"AL EMPEZAR LA SEMANA";WK:GOSUB 9200
4310 S$="TAMBIEN POSEES:*":GOSUB 9100
4320 PRINT:GOSUB 9200
4322 IF OA(1)=0 THEN 4332
4325 PRINT OA(1);:S$="FRASCOS DE MEDICINA*":GOSUB 9100
4330 GOSUB 9200
4332 IF OA(2)=0 THEN 4342
4335 PRINT OA(2);:S$="ARMAS*":GOSUB 9100
4340 GOSUB 9200
4342 IF OA(3)=0 THEN 4352
4345 PRINT OA(3);:S$="SACOS DE SAL*":GOSUB 9100
4350 GOSUB 9200
4352 IF OA(4)=0 THEN 4362
4355 PRINT OA(4);:S$="BALAS DE TELA*":GOSUB 9100
4360 GOSUB 9200
4362 IF OA(5)=0 THEN 4372
4365 PRINT OA(5);:S$="CUCHILLOS*":GOSUB 9100
4370 GOSUB 9200
4372 IF OA(6)=0THEN 4380
4375 PRINT OA(6);:S$="JOYAS*":GOSUB 9100
4380 GOSUB 9200:PRINT
4382 PRINT "TE QUEDAN";MO;:S$="  PIEZAS DE ORO*":GOSUB
     9100
4384 GOSUB 9200
4397 PRINT:S$=K$:GOSUB 9100
4398 GET I$:IF I$=""THEN 4398
4399 RETURN
```

Subrutina factura salarial

```
5000 REM COMPROBAR FACTURA SALARIAL
5005 IF MT>MO THEN 5010
5008 GOTO 5099
5010 PRINTCHR$(147)
5020 PRINT:PRINT:PRINT
5025 S$="ENTRE LA TRIPULACION CORRE EL RUMOR*":GOSUB
     9100
5030 S$="DE QUE NO TIENES SUFICIENTE*":GOSUB 9100
5035 S$="ORO PARA PAGARLES CUANDO TERMINE*":GOSUB
     9100
5040 S$="EL VIAJE.*":GOSUB 9100
5045 GOSUB 9200:PRINT
5050 S$="LOS ANIMOS SE ESTAN EXALTANDO !!*":GOSUB 9100
5055 GOSUB 9200:PRINT
5060 S$="ESPEREMOS QUE CONSIGAS*":GOSUB 9100

5065 S$="OBTENER ALGUN BENEFICIO!*:GOSUB 9100
5066 GOSUB 9200
5070 PRINT:S$=K$:GOSUB 9100
5080 GET I$:IF I$="" THEN 5080
5099 RETURN
```

Subrutina reparto de raciones

```
5100 REM REPARTIR RACIONES
5103 PRINTCHR$(147)
5105 S$="   REPARTIENDO LAS RACIONES*  ":GOSUB 9100
5106 S$="   _____________________*  ":GOSUB 9100
5107 GOSUB 9200:PRINT"SEMANA";WK:PRINT
5108 H$="N"
5110 FOR T=1 TO 4
5112 HR(T)=1
5115 IF PA(T)>0 THEN 5180
5120 PRINT"NO QUEDA  ";P$(T);"  !!!":GOSUB 9200
5130 S$="LA TRIPULACION SE ESTA DEBILITANDO !!*":GOSUB
     9100
5135 WK=10:GOSUB 9300
5139 GOTO 5290
5180 X=(PN(T)*CN)*(JL-WK+1)
5185 IF PA(T)<X THEN 5200
5190 GOTO 5270
5200 PRINT"QUEDA POCA  ";P$(T)
5205 GOSUB 9200
5210 S$="QUIERES PONER A LA TRIPULACION A*":GOSUB 9100
5215 PRINT"MEDIA RACION DE  ";P$(T)
5220 INPUT I$:I$=LEFT$(I$,1)
5221 IF I$<>"S" AND I$<>"N" THEN 5220:REM ERROR EN
     ENTRADA
5225 IF I$="N" THEN 5270
5230 HR(T)=.5:H$="S"
5240 WF=5:GOSUB 9300
5250 S$="LA TRIPULACION SE ESTA DEBILITANDO!*":GOSUB
     9100
5270 X=PN(T)*HR(T)*CN
5272 IF X>PA(T)THEN X=PA(T)
5275 PA(T)=PA(T)-X
5280 IF PA(T)=0 THEN PA(T)=-999
5285 PRINT X;U$(T);"S DE  ";P$(T);"  REPARTIDAS"
5290 PRINT:GOSUB 9200:NEXT
5295 PRINT:S$=K$:GOSUB 9100
5298 GET I$:IF I$="" THEN 5298
5299 RETURN
```

Reducción de la fortaleza de la tripulación

```
9300 REM REDUCIR FORTALEZA TRIPULACION EN FUNCION DE WF
9310 FOR S1=1 TO 16
9320 TS(S1,2)=TS(S1,2)-WF
9330 IF TS(S1,2)<1 THEN TS(S1,2)=-999
9340 NEXT
9399 RETURN
```



Gestor económico

El Sanyo MBC-550 representa una interesante opción en el mercado de ordenadores de 16 bits basados en el Intel 8088

A pesar de la rápida caída de los precios de los procesadores de 16 bits, la mayoría de los ordenadores basados en esta tecnología, con la excepción del Sinclair QL, siguen siendo muy caros. Las causas de este hecho son difíciles de comprender. Sinclair Research ha demostrado que es posible producir de forma rentable una máquina de 16 bits fijándole un precio razonable, y, sin embargo, la mayoría de los micros comparables tienen un precio muy superior al de ésta. La razón probablemente pueda atribuirse al tipo de cliente que existe en el mercado para tales máquinas. Éstos tienden a ser usuarios de gestión que, en opinión de los fabricantes, parecen estar en condiciones de pagar un poco más. Esta apreciación parece bastante acertada en el caso de los ordenadores basados en la serie de procesadores Intel 8088 y, probablemente, esté relacionada con la política de ventas del líder del mercado, IBM.

Pero todavía hay una parte del mercado que no está en condiciones de adquirir muchos de estos micros, aunque necesiten un ordenador con fines de gestión empresarial. Hasta hace muy poco, este sector del mercado estaba ocupado por ordenadores de oficina basados en el procesador Z80, equipados con unidades de disco que les permitían ejecutar CP/M. No obstante, actualmente muchos fabricantes están percatándose del potencial de este sector menoscabado del mercado de gestión y están volcándose en la producción de máquinas económicas basadas en el 8088.

El Sanyo MBC-550 es una de las primeras de éstas en salir al mercado. Aunque Sanyo no declare de forma manifiesta que sea compatible con el IBM PC, el MBC-550 ejecuta *de facto* el sistema de archivos en disco MS-DOS estándar. La máquina se vende con una única unidad de disco o bien en versión de unidades gemelas.

Al igual que la mayoría de las máquinas de oficina, el MBC-550 posee dos unidades separadas, el teclado y el ordenador principal. La carcasa de ambos es una combinación de plástico y metal con un acabado metálico plateado. El teclado está dividido en tres secciones. En el extremo izquierdo hay cinco teclas de función programables que, usadas junto con la tecla Shift, producen un máximo de 10 funciones que varían según la aplicación que se esté utilizando en cada momento. En BASIC, estas teclas se pueden emplear como entradas de palabras clave individuales, con las instrucciones utilizadas más comúnmente, como LIST, LOAD y RUN.

El teclado de máquina de escribir tiene un buen trazado, con todas las teclas de control en sus posiciones habituales. Las teclas de control incluyen un Backspace con borrado y una tecla Insert/Delete, siendo operativa una sola de las dos en cada aplicación. Hay, asimismo, una gran tecla Return, cuyo tamaño es cuatro veces el de las teclas normales. A la derecha de la barra espaciadora hay una tecla Graphics que, al trabarse, produce en la pantalla caracteres de gráficos. En el extremo derecho del teclado hay un teclado numérico que se puede utilizar como calculadora o, de pulsarse la tecla Number Lock, como control del cursor, para posibilitar la edición completa en pantalla en el BASIC.

El ordenador propiamente dicho es una gran caja plana del tamaño de un aparato de video, que se ha diseñado del tamaño necesario para colocar sobre ella la pantalla opcional Sanyo. Al igual que el teclado, está construido en su mayor parte con láminas metálicas. En la parte frontal del ordenador está el interruptor de potencia y hay espacio para dos unidades de disco, si bien en nuestra fotografía vemos el modelo de una sola unidad. Las unidades que emplea el MBC-550 son del tipo estándar de discos flexibles de 5 ¼ pulgadas. En la parte delantera de cada unidad hay un clip para mantener el disco en posición y evitar que pueda ser sacado mientras la cabeza de la unidad lo está leyendo.

Diferencias fraternales
El Sanyo MBC-550 es una máquina de oficina de 16 bits que se vende a un buen precio y opera bajo el popular sistema operativo MS-DOS. La máquina viene en dos versiones: la MBC-550 es la versión de una sola unidad de disco; la versión que configura la doble unidad es la denominada MBC-555. La pantalla que vemos en la fotografía no está incluida en el precio básico



Chris Stevens

Interfaces para periféricos

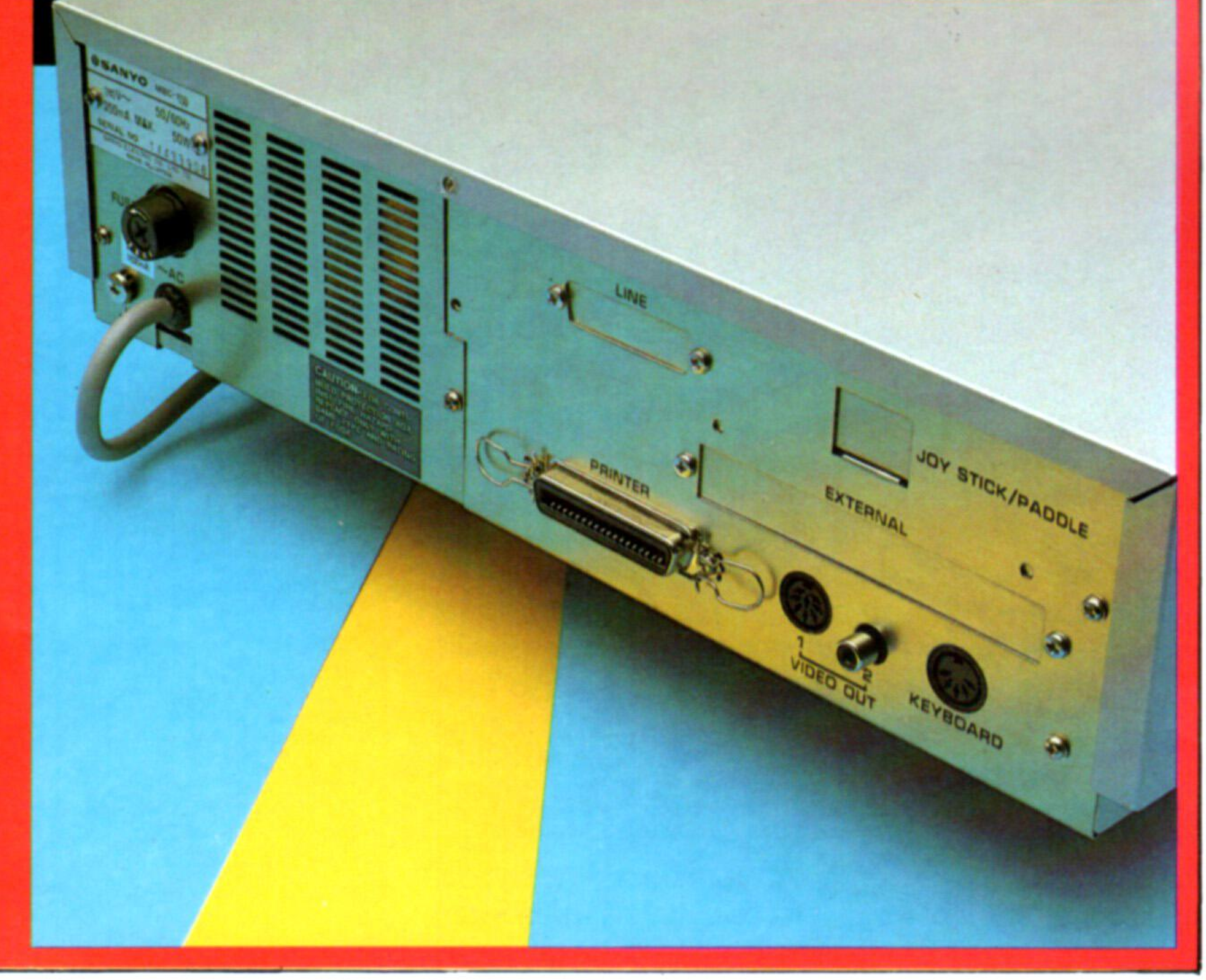
En su parte posterior, el ordenador contiene la fuente de alimentación eléctrica y las interfaces para periféricos. En el lado izquierdo de su parte

posterior se halla el cable de potencia, la caja del fusible y la conexión a tierra. A la derecha hay una interface en paralelo Centronics que permite la conexión de una impresora. A la derecha de ésta hay un par de conectores para pantalla. El primero es un conector RGB que permite que el ordenador haga funcionar una pantalla en color, mientras que el otro es un enchufe de video compuesto para la pantalla monocromática verde de Sanyo. Encima de estas interfaces hay otras puertas de ampliación que en el futuro darán cabida a interfaces extras para periféricos.

La puerta Line de encima de la interface para impresora se incluye para la conexión de una interface en serie RS232 que facilitaría la comunicación del ordenador con otras máquinas, a través de un modem, o con cualquier otro dispositivo en serie, como una impresora. Junto a ella hay una puerta en la que se puede instalar una interface para palanca de mando de la serie MBC o bien compatible con Atari. Ésta no sólo permite la instalación de una palanca de mando sino que también permitiría controlar el ordenador mediante un mando de bola, un mando de raqueta (*paddle*) u otro dispositivo externo. Hay, asimismo, un conector para bus externo que permite la conexión en interface de otros dispositivos periféricos.

A diferencia de muchos fabricantes que se oponen taxativamente a que los usuarios manipulen sus máquinas, en el manual para el usuario Sanyo ha incluido esmeradamente instrucciones para la instalación de todas estas interfaces para periféricos, bancos extras de RAM y una segunda unidad de disco por si hubiera instalada sólo una. Esta inclusión es muy de agradecer, aunque la guía menciona que, en caso de alguna duda, los usuarios deberían consultar con un ingeniero cualificado. El resto del manual para el usuario es igualmente valioso. Contiene un glosario de términos de informática, una descripción completa del BASIC Sanyo, incluyendo instrucciones para utilizar el sistema de disco desde BASIC, y gran cantidad de información técnica.

Espacio para ampliaciones
Aunque el MBC-550 no esté equipado con tantas opciones para periféricos como algunas de las máquinas más caras que existen en el mercado, Sanyo la ha diseñado para que sea ampliable. Hay ranuras extras para interfaces para palancas de mando y dispositivos en serie y en el manual para el usuario se ofrecen instrucciones para instalarlas

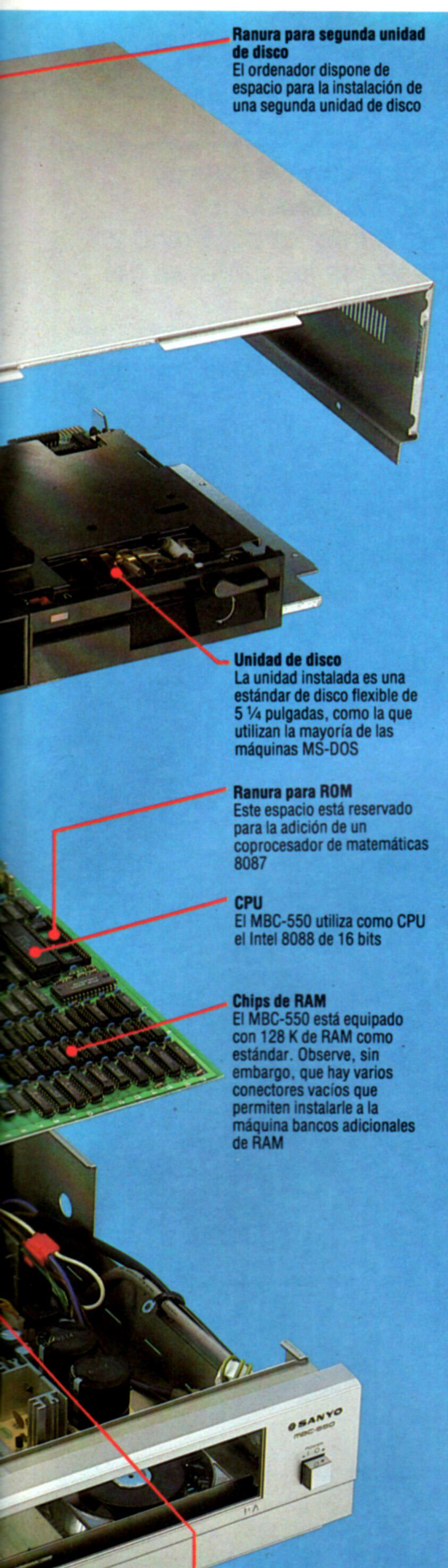


Interface para impresora
Esta puerta permite conectar cualquier impresora en paralelo compatible con Centronics

Chip de interface para periféricos
El chip 8255A controla los diversos periféricos que se pueden conectar

Chip video
Este chip permite utilizar cualquier sección de la RAM de 256 K para almacenamiento en RAM de video

Altavoz
La capacidad de sonido del ordenador se canaliza a través de este altavoz incorporado



Ranura para segunda unidad de disco
El ordenador dispone de espacio para la instalación de una segunda unidad de disco

Unidad de disco
La unidad instalada es una estándar de disco flexible de 5 ¼ pulgadas, como la que utilizan la mayoría de las máquinas MS-DOS

Ranura para ROM
Este espacio está reservado para la adición de un coprocesador de matemáticas 8087

CPU
El MBC-550 utiliza como CPU el Intel 8088 de 16 bits

Chips de RAM
El MBC-550 está equipado con 128 K de RAM como estándar. Observe, sin embargo, que hay varios conectores vacíos que permiten instalarle a la máquina bancos adicionales de RAM

Fuente de alimentación
El MBC-550 posee su propio transformador de potencia a bordo, instalado en la parte trasera

El BASIC Sanyo, un derivado de la versión de Microsoft, es ciertamente adecuado, aunque parece ser algo lento para procesar funciones aritméticas. Ya se ha notado desde hace mucho tiempo que el chip 8088 es perezoso para procesar matemáticas (a ello se debe la popularidad del coprocesador de matemáticas 8087), pero el MBC parece aún más lento que la mayoría de las otras máquinas basadas en el 8088. Del mismo modo, el dibujo de líneas en el MBC-550 es también muy lento. Esto se debe a que el BASIC Sanyo carece de una instrucción DRAW, lo que exige llevar a cabo el trazado de líneas mediante la instrucción PSET, que simplemente enciende un pixel especificado con un color dado. Por lo tanto, el trazado de una línea supone utilizar un bucle.

Aparte de instrucciones estándares tipo Microsoft, tales como MID$, LLIST y CIRCLE, también hay instrucciones para definir y contemplar ventanas en la pantalla. Se fomenta la programación estructurada mediante la inclusión de la sentencia condicional WHILE...WEND. Muchas palabras clave del BASIC Sanyo se pueden entrar mediante dos o tres pulsaciones de tecla basadas en la tecla Control. Por ejemplo, la palabra clave DIM se puede entrar pulsando simultáneamente CTRL, SHIFT y D, mientras que PRINT se entra pulsando CTRL y P. Si bien todo intento por simplificar la entrada de palabras clave del BASIC es digno de elogio, y muchas de las instrucciones con pulsaciones de tecla abreviadas guardan alguna relación con los originales, es difícil imaginarse que alguien intente siquiera recordar las 40 instrucciones disponibles para escribir programas en BASIC. Lo más probable es que intente memorizar algunas de las versiones abreviadas empleadas más comúnmente.

Con el Sanyo MBC-550 viene un disco de sistema MS-DOS, que contiene también el BASIC Sanyo, el paquete para tratamiento de textos *WordStar*, tan ampliamente utilizado, y la hoja electrónica *CalcStar*. Estos paquetes van acompañados de un manual que ofrece descripciones completas para su empleo.

Cabría esperar que el MBC-550 estándar (basado en el chip 8088 y operando bajo MS-DOS) tuviera pocos problemas para ejecutar la ingente cantidad de software IBM PC que existe en el mercado. Pero lo cierto es que las restricciones de hardware de la máquina significan que ninguno de los paquetes compatibles con IBM que probamos en el equipo se pudo ejecutar.

Sin ninguna duda, existe un mercado para máquinas de oficina MS-DOS de precio reducido capaces de ejecutar software compatible con IBM. No se trata ya de esperar que aparezcan, sino de cuándo aparecerán. El Sanyo MBC-550 es un primer intento, pero, lamentablemente, su falta de compatibilidad lo dejará en una situación de aislamiento.

Para quien sea propietario de una pequeña empresa que sólo necesita un ordenador para ejecutar facilidades competentes de tratamiento de textos y hoja electrónica, el MBC-550 parece una buena propuesta.

No obstante, quien desee tener acceso a una base de software más amplia de la que hay disponible para el Sanyo MBC-550, deberá resignarse ya sea a gastar algo más o bien a esperar un poco más de tiempo.

SANYO MBC-550

DIMENSIONES
375×355×108 mm

CPU
Intel 8088 operando a 3,6 MHz

PANTALLA
Pantalla para texto de 80×25, dando 640×200 pixels en modalidad de alta resolución. En alta resolución hay hasta ocho colores disponibles

INTERFACES
Puerta en paralelo Centronics, con opciones para la instalación de una interface RS232, y una puerta para palanca de mando compatible con Atari

LENGUAJES DISPONIBLES
BASIC Sanyo

TECLADO
65 teclas tipo máquina de escribir, cinco teclas de función doble programables y un teclado numérico de 19 teclas

DOCUMENTACION
El manual es muy bueno y contiene una explicación completa de las palabras clave del BASIC, aunque, lamentablemente, no contiene un curso de aprendizaje. Asimismo, hay una introducción al MS-DOS, *WordStar* y *CalcStar*. Insólitamente, tratándose de una máquina de oficina, el manual para el usuario proporciona gran cantidad de información técnica, incluyendo instrucciones que le indican al usuario cómo instalar sus propias placas de ampliación

VENTAJAS
Por su precio, el Sanyo MBC-550 es, ciertamente, una buena compra, proporcionando una informática de gestión completa de 16 bits a una fracción del precio normal

DESVENTAJAS
Debido a que la máquina carece de compatibilidad con la mayor parte del software existente, el ordenador se encuentra con el problema habitual de las máquinas nuevas: la falta de programas disponibles

Ensamblaje del brazo

Ensamblaremos el cuerpo principal, las secciones superior e inferior del brazo y efectuaremos las junturas del codo y el hombro

Paso 1: Cuerpo principal

PIEZA LATERAL
PIEZA CRUZADA DE 6 MM
TUBO DE BRONCE DE 6 MM
BASE SUPERIOR
COJINETE
BASE INFERIOR
ESPACIADOR
TARUGO DE ACERO
VISTA AMPLIADA

CUERPO PRINCIPAL
PIEZA CRUZADA DE 6 MM
TUBO DE BRONCE DE 6 MM
PIEZA LATERAL
BASE SUPERIOR
BASE INFERIOR
COJINETE REBORDEADO
ESPACIADOR
TARUGO DE ACERO
CAJA BASE
ELEVACION LATERAL

Paso 1: Cuerpo principal

Taladre un agujero en la pieza inferior de la base (la más ancha de las dos piezas) para aceptar el cojinete. Utilizando una broca de mayor tamaño o un avellanador, bisele el labio superior del agujero para asentar en él el reborde del cojinete, como se ve en la ilustración. Después taladre un agujero de 6 mm de diámetro en la pieza de la base superior para recibir el trozo de 25 mm de tubo de 6 mm de diámetro interno. Con los agujeros alineados y el cojinete asentado sobre la pieza inferior de la base, pegue las dos piezas entre sí. Inserte el tubo de bronce en el agujero, desde arriba, y péguelo en su sitio. Cuando este ensamblaje esté seco se pueden añadir las piezas laterales del cuerpo. Perfore los agujeros en la parte de arriba de las piezas laterales de modo que acepten estrechamente un trozo de tubo de 4 mm de diámetro externo: se trata de un ajuste de fricción. Inserte un trozo de madera de 6×6 mm y 37 mm de largo a modo de pieza cruzada entre los dos laterales (colocada en la parte trasera del ensamblaje, en línea con el corte para el motor de la pieza lateral) y pegue el ensamblaje. Cuando éste esté seco, coloque un espaciador (p. ej., una tuerca grande) en el tarugo de acero que sobresale desde la parte superior de la caja base ensamblada anteriormente. Empuje el tarugo a través del cojinete y compruebe que el ensamblaje del cuerpo principal pueda girar libremente.

Paso 2: Brazo inferior

Corte un trozo de 37 mm de tubo de bronce de 5 mm de diámetro externo y taladre agujeros en el medio de las piezas del brazo inferior para recibir este tubo. Asegúrese de obtener un ajuste apretado, puesto que este tubo de bronce formará parte de la juntura del hombro. Perfore otros dos agujeros en el extremo piramidal de las piezas del brazo inferior para recibir el tubo de bronce de 4 mm de diámetro externo. Como antes, asegúrese de conseguir un ajuste de fricción. Trabajando en el medio, pase el tubo de 5 mm a través de los agujeros de las piezas del brazo inferior utilizando, como vemos en la ilustración, uno de los servomotores como espaciador. Corte dos trozos de 20 mm de largo de madera de 6 mm como piezas cruzadas y colóquelas entre las piezas del brazo inferior. Éstas van justo debajo del tubo de bronce y en la parte de abajo del ensamblaje, y se emplean para montar el servomotor. Péguelos en su sitio y, una vez secos, monte el motor utilizando cuatro tornillos sin tuerca y arandelas de goma. Ya se puede unir el ensamblaje del brazo inferior al cuerpo principal, por la juntura del hombro. Ahora inserte un trozo de 45 mm de largo de tubo de 4 mm a través de uno de los agujeros de la parte superior del cuerpo principal y a través del tubo de bronce que ya estaba colocado en el ensamblaje del brazo inferior. Asegúrese de que el ensamblaje del brazo inferior esté centrado. Marque la posición en la que parezca estar correctamente alineado, desmóntelo, y luego pegue el tubo de bronce más grande en su lugar en el ensamblaje del brazo inferior. Una vez seco, vuelva a ensamblar la juntura del hombro.

Paso 3: Brazo superior

Perfore dos agujeros para recibir el tubo de bronce de 5 mm en los extremos más anchos de las piezas del brazo superior. Perfore otros dos agujeros en los otros extremos. Pegue un trozo de 25 mm de madera de 6×6 mm en la pieza en "T" y moldee los extremos. Éste actuará a modo de espaciador de la muñeca. Coloque la sección en "T" entre las dos piezas del brazo superior y taladre a través del espaciador de la muñeca de modo que a través del espaciador y de las secciones del brazo superiores se pueda pasar un tornillo para metales. Fije la pieza en "T" en su sitio con una arandela y un tornillo. En el otro extremo del brazo, empuje a través de los agujeros taladrados un trozo de 17 mm de tubo de bronce de 5 mm, y conecte el ensamblaje del brazo superior al brazo inferior en la juntura del codo pasando a través de la juntura, como antes, un trozo de 25 mm de tubo de 4 mm. Compruebe que el brazo superior esté centrado antes de desmontarlo y pegar en su sitio el tubo de bronce de diámetro mayor.



Paso 2: Brazo inferior

Paso 3: Brazo superior

Kevin Jones

Conjuntos

Veamos cómo se utilizan los conjuntos en PASCAL y cuál es el orden de precedencia de los operadores de conjuntos

Además de sus estructuras de datos típicas, tales como matrices y archivos, el PASCAL incluye conjuntos y registros. Examinemos los primeros.

Con frecuencia hablamos del *juego* o *conjunto* de caracteres de un ordenador. Pero, ¿qué es precisamente un *conjunto* y en qué se diferencia de una matriz? La principal característica de un conjunto reside en que es un grupo de objetos que se pueden procesar como una única entidad, en vez de tener que acceder a cada elemento de forma individual. Un ejemplo práctico sería el conjunto de todas las personas que programan en PASCAL, o el de las letras del alfabeto que son vocales. Con frecuencia no hay implicado ningún orden particular, siendo la única pregunta de interés: ¿es o no es este objeto determinado un miembro del conjunto?

Por razones de eficacia de implementación, el PASCAL impone algunas restricciones a los conjuntos. Sólo pueden tener tipos escalares simples como miembros (no matrices ni otros datos estructurados) y hay un límite máximo (definido por implementación) para la gama permitida para este tipo "base". La sintaxis es directa; por ejemplo:

```
TYPE
  Numeros     =SET OF 0..127;
  Alfabeto    =SET OF 'A'..'Z';
  MezclaCol   =SET OF (Rojo,Verde,Azul);
```

La gama de los valores posibles del tipo base ordinal del conjunto se puede expresar con la misma sintaxis que para tipos de subrango.

Las variables de conjunto se declaran en la parte de declaración VAR, de la misma forma que se declaran todas las variables en PASCAL, y pueden aparecer en sentencias literalmente (es decir, como "literales") encerradas entre corchetes:

```
VAR
  codigos   : Numeros;
  paleta    : MezclaCol;
BEGIN
  codigos   :=[0..2,4,8,16,32,64];
  paleta    :=[Rojo..Azul];{etc.}
```

Este segmento inicializaría los códigos de conjunto de modo que contuvieran sólo los números del 0 al 2 inclusive y 4,8, etc., tal como se listan. Dado que en el conjunto no hay ningún orden inherente (al contrario que en su tipo base), podrían igualmente expresar este conjunto como [64,32,16,8,4,0..2], pero observe que el subrango 2..0 (ilegal en una verdadera definición de subrango) indicaría meramente un rango vacío. De hecho, se puede inicializar como conjunto vacío cualquier conjunto, mediante la sentencia: Conjunto1:=[], que da origen a la única excepción a la regla general del PASCAL, a saber: el tipo de cualquier literal se conoce mediante la inspección. Sin que al menos un miembro del conjunto aparezca de forma literal, ni nosotros ni el ordenador podemos determinar su tipo. Afortunadamente, el conjunto vacío sólo puede ocurrir en asignaciones (como en el ejemplo) o en expresiones en las cuales los otros identificadores ya tendrán declarados sus tipos. Ello significa, sin embargo, que el conjunto vacío es un subconjunto de todos los tipos de conjuntos, pero eso es natural.

Uno de los operadores más útiles que proporciona el PASCAL para estructuras de conjuntos es, como DIV y MOD, una palabra reservada: IN. Nos permite comprobar la pertenencia a un conjunto y es un operador de relación que toma dos operandos. El lado izquierdo debe ser una expresión que evalúe uno de los posibles miembros del conjunto (en otras palabras, un valor del tipo base del conjunto) y el lado derecho puede ser una variable de conjunto o un literal de conjunto. La expresión dará un resultado booleano: verdadero si el valor pertenece al conjunto y falso en caso contrario.

Por consiguiente: N IN codigos y Verde IN paleta son expresiones booleanas legales. La comprobación habitual de si un carácter es un dígito se podría escribir así:

```
IF(c>='0')AND(c<='9')THEN...
```

Mucho menos confuso sería comprobar si el valor de c pertenece al conjunto de caracteres en el cual estamos interesados, utilizando:

```
IF c IN('0'..'9')THEN...
```

O, quizá, si estamos escribiendo un programa para un juego de naipes:

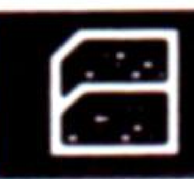
```
TYPE
  rango=(dos,tres,cuatro,cinco,seis,siete,ocho,
         nueve,diez,Valet,Dama,Rey,As);
  ConjNaipes=SET OF rango;
VAR
  naipe : rango;
  figuras : ConjNaipes;
BEGIN
  figuras:=(Valet..As);
  IF naipe IN figuras THEN {etc.}
```

Compare este último ejemplo con:

```
IF(naipe=Valet)OR(naipe=Dama)OR(naipe=Rey)
OR...
```

En PASCAL también hay operaciones definidas sobre conjuntos como un todo. Éstas son intersección, unión y diferencia. Si B es el conjunto de todos los programadores de BASIC y P representa a los de PASCAL, la intersección de ambos conjuntos es el conjunto de programadores que utilizan el BASIC y el PASCAL *a la vez*. La unión de P y B es el conjunto de personas que programan o en PASCAL o en BASIC, es decir, la *combinación* de los dos conjuntos.

La diferencia de conjuntos es, como su nombre sugiere, el resultado de suprimir o restar todos los miembros de un conjunto del otro. Por consiguiente, en la notación del PASCAL, P−B representa a



Snooker

El snooker (billar inglés) se juega con 15 bolas rojas (todas las cuales valen un punto si se las hace caer en un agujero), una bola "pinta" blanca y los seis "colores", cuyos puntos son los siguientes:

amarillo	2
verde	3
marrón	4
azul	5
rosa	6
negro	7

Tras hacer caer cada una de las bolas rojas, se le permite al jugador intentar colocar una de las bolas de color. Los colores de las bolas colocadas se quitan luego de la tronera y se vuelven a colocar en la mesa. Tras la última combinación de rojo y color, se deben colocar todos los colores por orden ascendente de valor. En el snooker, un *break* es una serie de golpes legales que terminan ya sea en el fracaso de un jugador en colocar una bola, en un golpe fallado o en el final del juego (cuando ya no quedan más bolas). Escriba un programa para calcular el máximo *break* posible, pero (y aquí está la trampa) asegurándose de que el único número que aparezca en el programa sea el 15. La solución, en el próximo capítulo

Ian McKinnell

Precedencia de operadores

En la tabla de abajo vemos el orden de precedencia de los operadores. El paréntesis obliga a evaluar por separado la expresión encerrada, rompiendo el orden normal de precedencia (que se refleja aquí) si fuera necesario

Precedencia	Operadores	Tipo
Máxima	NOT + −	{unarios}
	AND */ DIV MOD	{multiplicación}
	OR + −	{adición}
Mínima	<<=<>>=>IN	{de relación}

todas las personas que programan en PASCAL pero no en BASIC. Del mismo modo, la notación que se emplea para uniones es P+B, y para la intersección P*B. Estos símbolos de operadores resultan ser los mismos que los bien conocidos operadores aritméticos, pero no se debe confundir la distinta clase de operaciones. En realidad, la razón por la que aparecen con tanta naturalidad en el contexto de las operaciones con conjuntos, es porque representan la comprobación de bits involucrada.

Tomemos un conjunto de ocho elementos. La presencia de un miembro determinado se podría indicar estableciendo el bit apropiado de un patrón de 8 bits (un byte); la ausencia en el conjunto se podría señalar, del mismo modo, mediante un cero. Las comprobaciones de pertenencia, entonces, requieren apenas una máscara o comprobación de bit, una unión de conjuntos se convierte en una operación OR y la intersección en un simple AND. Estas operaciones están invariablemente disponibles a nivel de lenguaje máquina, de modo que el compilador de PASCAL puede proporcionar estructuras de datos de conjuntos y las operaciones con ellas relacionadas con gran eficacia. El gasto de memoria es asimismo mínimo y, especialmente cuando el tamaño de los conjuntos se puede mantener dentro del tamaño de palabra del ordenador (en máquinas de 32 bits y más, por ejemplo), las operaciones con conjuntos pueden ser las que se implementan con mayor eficiencia en PASCAL.

Estamos ahora en condiciones de resumir todos los operadores del PASCAL. Aquellos de los que no hemos hablado de forma explícita son ya familiares en otros lenguajes, y el PASCAL facilita mucho las cosas al tener sólo cuatro niveles de precedencia de operadores. Naturalmente, los operadores *unarios* o *monádicos* son los que tienen precedencia sobre todos los demás. Éstos son los símbolos + y −, que indican el signo de un número, y el operador de negación booleana NOT. En el segundo nivel de precedencia se hallan todos los operadores de "multiplicación" (incluyendo los signos de división) seguidos por la suma/resta y, en el nivel inferior de precedencia, todos los operadores de relación, incluyendo IN.

Observe que los otros dos operadores booleanos (AND y OR) se tratan de forma correcta como operadores de multiplicación y adición, respectivamente. Ello evoca de manera fidedigna la verdadera álgebra booleana implicada, y significa que muchas condiciones de relación se encierran entre paréntesis para anular esta precedencia. De lo contrario, por ejemplo: IF N>0 AND N<10 THEN.. dará un error en tiempo de compilación, porque la expresión 0 AND N (que se evaluaría primero) intenta combinar dos operandos enteros con un operador booleano.

Los operadores del mismo nivel de precedencia se evalúan de izquierda a derecha, como es habitual. El símbolo utilizado para la asignación (:=) tiene una precedencia inferior a cualquiera de los operadores anteriores, porque la expresión del lado derecho de una asignación debe haber sido evaluada completamente antes de efectuar la asignación. Un último aviso: uno jamás puede dar por sentado que no se evaluará alguna parte de una expresión, de modo que: IF(N>0)AND(K/N<10) THEN... podría colgar el programa si N fuera cero (¡lo que crearía un error de división por cero!).

¡Bingo!

Un cartón de bingo se puede representar mediante un conjunto. Aunque los elementos base (enteros comprendidos entre el 1 y el 90) están ordenados, la única consideración vital es si el número cantado está o no en el cartón. Al decir del PASCAL: "numero IN Carton" es o bien verdadero, o falso. El programa simula una partida de bingo leyendo primero los números del cartón desde el teclado y luego comprobando si el conjunto de los números cantados es un subconjunto del cartón. La expresión Carton-Cantado se convertirá en el conjunto vacío cuando todos los miembros de Carton estén también en el conjunto denominado Cantado. La adición de un miembro a un conjunto que ya lo contenga no provoca ninguna modificación en el conjunto, así como "quitar" un no miembro. Observe que no podemos añadir directamente un miembro a un conjunto; pero creado un conjunto de un solo elemento, encerrando el número entre corchetes, podemos obtener la unión de dos conjuntos. Tal como está, el programa permitirá entradas de cartón duplicadas, y aceptará un número ilegal fuera de la escala 1-90, produciendo un error de ejecución. A modo de ejercicio, piense en alguna forma sencilla de añadir construcciones de bucle para evitar estas anomalías y rechazar los números ya cantados

```
PROGRAM          Bingo      (input,output);

CONST
        Columnas     =  40;     {segun medida VDU}
        Medio        =  25;     {segun medida VDU}

TYPE
        Bingo        =  SET OF 1..90;

VAR
        contador     :  1..15;
        Reconocido,
        Cantado,
        Vacio,
        Carton       :  Bingo;
        numero       :  integer;
        Casa         :  boolean;

BEGIN
   Vacio :={  };
   Reconocido :=[1..90];
   WriteLn ('***BINGO***' : Medio);
   WriteLn;
   WriteLn ('Entre los 15 numeros del carton,'  );
   WriteLn ('(un RETURN tras cada uno de ellos) :');
   Carton := Vacio;

FOR contador :=1 TO 15 DO
   BEGIN
      write (contador : 10, ': ? ');
      ReadLn (numero);
      Carton :=Carton+[numero]
   END;

   WriteLn;
   WriteLn ('PREPARADO !' : Medio);
   WriteLn;
   WriteLn ('Ahora cante cada numero :');
   Cantado := Vacio;

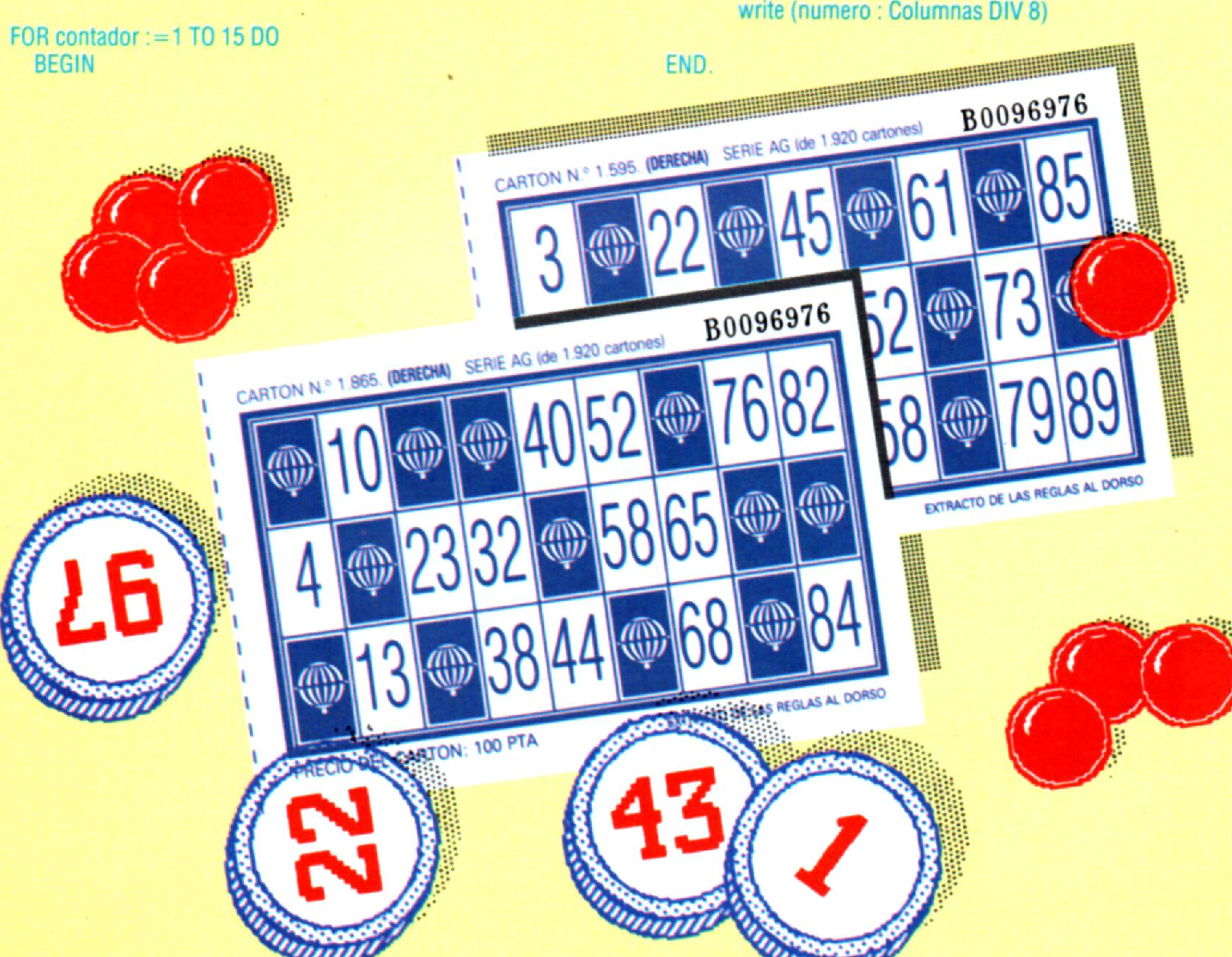
   REPEAT
      write ('?': Medio);
      ReadLn (numero);
      Cantado :=Cantado+[numero];
      Casa := Carton - Cantado = Vacio

   UNTIL Casa;

   WriteLn;
   WriteLn ('Felicitaciones !' : Medio);
   WriteLn ('Sus numeros fueron:');
   WriteLn;

   FOR numero :=1 TO 90 DO
      IF numero IN Carton THEN
         write (numero : Columnas DIV 8)

END.
```





Última llamada

Examinadas ya las rutinas más importantes que componen el OS del BBC Micro, concluiremos el estudio con algunos programas que ilustran los usos de las llamadas del OS y dejan en evidencia algunas de sus limitaciones

En el capítulo anterior dimos un programa como ejercicio que muestra muy bien cómo podemos interceptar uno de los vectores del OS para cambiar el modo como responde éste a ciertos eventos. El programa altera el contenido de OSWRCHV, el vector que contiene la dirección de la rutina OSWRCHV. Este vector se encuentra en las direcciones &20E y &20F.

Una de las funciones de OSWRCH es tomar el texto en BASIC e imprimirlo en la pantalla cuando se lista un programa. La principal función del código máquina en nuestro programa es examinar el código ASCII de cada carácter según pasa por el acumulador durante la rutina OSWRCH. Si el código ASCII está entre el 64 y el 91 (correspondiente a las mayúsculas), entonces se agrega 32 al valor para dar su equivalente en minúsculas. De esta manera el programa forma una "cuña" que se ejecutará cada vez que el sistema operativo llame a OSWRCH. Después de que se ha ejecutado nuestro código, se pasa el control a la rutina real OSWRCH por medio de la instrucción JMP con el anterior contenido de OSWRCHV. Después de ejecutar el programa, todas las mayúsculas del programa listadas en la pantalla o digitadas en el teclado se convertirán en minúsculas por obra de nuestra rutina. Al pulsar BREAK, se restablece a OSWRCHV en sus valores normales.

Programa de empleo de memoria

Este listado es otro programa de utilidad, que inspecciona el empleo de la memoria mientras usted está programando.

La expresión:

HIMEM−(?2+256*?3)

proporciona la cantidad de memoria que queda después de ocupar su espacio el programa y las variables (las posiciones 2 y 3 contienen la dirección del límite superior del cuadro de variables en BASIC). Así la expresión indica cuánta memoria le queda disponible. Pero imprimir (PRINT) una y otra vez este valor sería algo tedioso. La rutina usa el evento Un carácter entra en el buffer de entrada para evaluar la expresión cada vez que se pulsa una tecla, y emite un pitido cuando la memoria restante para el BASIC desciende por debajo de un cierto nivel.

El programa se dispara por medio de un evento de pulsación de tecla y continuará operando entre bastidores mientras usted entra su programa.

La primera tarea del programa es determinar el nivel al cual el usuario desea ser advertido de la memoria que le queda. Esto lo hace PROCselect-memory, que establece el nivel en la forma *hi-lo*. El programa se posiciona en la dirección &0A00, y (como siempre ocurre con programas que emplean eventos) primero guarda el contenido de todos los registros llevándolos a la pila. Después de llamar la subrutina que se encarga de ello, los registros se restauran y se ejecuta RTS para devolver el control del programa allí donde estaba cuando sucedió el evento.

La principal subrutina llama a otras dos subrutinas para calcular la cantidad de memoria que queda. La rutina inc toma el valor almacenado en las posiciones 2 y 3 y le suma el byte *lo* y el byte *hi* del nivel, almacenando el resultado en las direcciones &70 y &71. La rutina sub realiza entonces un cálculo de 16 bits, restando el contenido de &70 y

Uso de la memoria

```
100 REM **** Empleo de evento por pulsacion de tecla ****
110 REM **** para alertar sobre el uso de la memoria ****
120 MODE 1
130 PROCselect_memory
140 PROCassemble
150 END
160 .
170 DEFPROCselect_memory
180 INPUT "Pitido despues de cuantos K restantes",n : n=n*1024
190 hibyte=n DIV 256
200 lobyte=n MOD 256
210 ENDPROC
220
230 DEFPROCassemble
240 oswrch=&FFEE
250
260
270 FOR pass=0 TO 3 STEP 3
280 P%=&0A00
290
300 [OPT pass
310 .start
320 PHP
330 PHA
340 TXA:PHA
350 TYA:PHA
360
370 JSR memory_check
380
390
400 PLA:TAY
410 PLA:TAX
420 PLA
430 PLP
440 RTS
450
460 .memory_check
470 JSR inc
480 JSR sub
490 BPL room
500 LDA #7
510 JSR oswrch
520 .room RTS
530
540 .sub SEC
550 LDA &06:SBC &70:STA &70
560 LDA &07:SBC &71:STA &71
570 RTS
580 .inc CLC
590 LDA &02:ADC #lobyte:STA &70
600 LDA &03:ADC #hibyte:STA &71
610 RTS
620
630 ]:NEXT
640 ?&220=start MOD 256
650 ?&221=start DIV 256
660 *FX14,2
670 ENDPROC
```

&71 de la HIMEM. El valor de HIMEM se almacena en la página cero en las posiciones &06 y &07. Obsérvese aquí cómo se han empleado posiciones de memoria absolutas, principalmente porque estas posiciones han recibido este empleo en todas las versiones del BASIC del BBC.

Una vez efectuada la resta, la línea 490 comprueba el signo del resultado. Si es positivo, indica que hay todavía más bytes de memoria que el nivel previamente determinado. Si es negativo, ya no queda memoria, por lo que se ha de emitir un pitido. La llamada OSWRCH con A=7 es la que se encarga de emitirlo.

Los dos programas hasta aquí analizados son sólo unos sencillos ejemplos de posibles y útiles aplicaciones. Una ventaja del uso de las llamadas del sistema operativo directamente es que podemos producir un código dirigido con eventos, facilidad imposible en BASIC.

Los dos programas que transcribimos a continuación emplean eventos.



Música de fondo

```
10 REM **** EVENTOS SONOROS ****
50
60 PROCassemble
80 CALL initialise
90 END
100
110 DEFPROCassemble
120 note_count=&70
130 oswrch=&FFEE
140 osbyte=&FFF4
150 osword=&FFF1
160
170 FOR pass=0 TO 2 STEP 2
180 P%=&0C00
190
200 [OPT pass
210 .initialise
220 LDA #event MOD 256:STA &220
230 LDA #event DIV 256:STA &221
240 LDA #14:LDX #5:JSR osbyte
250 LDA #0:STA note_count
260 JSR clock
270 RTS
280
290 .event
300 PHP
310 PHA
320 TXA:PHA
330 TYA:PHA
340
350 JSR play_note
360 JSR clock
370
380
390 PLA:TAY
400 PLA:TAX
410 PLA
420 PLP
430 RTS
440
450 .play_note
460 LDY note_count
470 .sloop LDA notetable,Y
480 STA soundtable+4
490 JSR sound
500 INC note_count
510 LDA note_count:CPM#5:BNE out
520 LDA #0:STA note_count
530 .out RTS
540
550
560 .clock
570 LDX #time MOD 256
580 LDY #time DIV 256
590 LDA #4
600 JSR osword
610 RTS
620
630 .sound PHA
640 TXA:PHA
650 TYA:PHA
660 LDX #soundtable MOD 256
670 LDY #soundtable DIV 256
680 LDA #7
690 JSR osword
700 PLA:TAY
710 PLA:TAX
720 PLA
730 RTS
740
750 .notetable
760 EQUD &00000000
770 EQUD &00000000
780 .soundtable
790 EQUD &00000000
800 EQUD &00000000
810 .time EQUD &FFFFFFCF
820 EQUB &FF
830 ]:NEXT
840 FOR I%=0 TO 4:READ data:?
    (notetable+I%)=data:NEXT
850 FOR I%=0 TO 7:READ data:?
    (soundtable+I%)=data:NEXT
860 ENDPROC
870 DATA 69,73,81,89,97
880 DATA 1,0,&FA,&FF,0,0,10,0
```

Este programa emplea el evento El reloj de intervalos se agotó para hacer sonar una frase de notas musicales mientras el ordenador continúa realizando otras tareas. Dado que la rutina del evento opera como fondo, se puede guardar el programa, editarlo, listarlo o ejecutar otro programa diferente, mientras suenan a cada momento las notas musicales. Se puede ampliar el programa para que suene una melodía mientras se ejecuta, por ejemplo, un programa de juego. La melodía se hará oír incluso durante operaciones con disco.

El programa se ensambló para que se inicie en la dirección &0C00. La primera sección es una rutina de inicialización que realiza las siguientes tareas. El vector de eventos EVENTV, en las direcciones &220 y &221, se cambia para que apunte a nuestra rutina. Se llama entonces a OSBYTE 14 con X=5 para activar el evento El reloj de intervalos se agotó. Un *contador de notas* se inicializa con valor cero en &70 y finalmente el reloj del evento es puesto en marcha con una llamada a la subrutina del reloj.

La primera tarea de la rutina de tratamiento de eventos es guardar en la pila los registros de la CPU. Seguidamente se llama una subrutina que hace sonar una sola nota y se restablece el reloj. Finalmente, se restauran los registros. Nótese que el fragmento del programa que guarda y restaura los registros es común a todas nuestras rutinas de tratamiento de eventos, y basta con cambiar el código entre estos dos bloques de instrucciones.

Examinemos las diferentes partes de la subrutina que hace sonar las notas. Cada vez que el reloj del evento pasa por el cero, suena una única nota, seleccionada de un cuadro de notas. El cuadro de notas, que llamamos notetable, consiste sencillamente en una serie de bytes que contienen la información de la altura de las cinco notas que deseamos tocar. Se emplea un contador para determinar cuál de las cinco notas ha de sonar, reteniendo el valor en &70. El valor de la altura se recoge y almacena en el lugar apropiado dentro de soundtable, que no es más que un bloque de parámetros para la rutina OSWORD que utilizaremos más tarde para producir el sonido. El contador se incrementa entonces, apuntando así a la siguiente posición dentro del cuadro de notas para la siguiente vez que entre en la rutina. Dado que se dispone de tan sólo cinco notas, tan pronto como la posición &70 retiene el valor cinco, el citado cuadro se restablece a cero.

La subrutina clock (reloj) utiliza OSWORD con A=4 para poner en acción el reloj del evento. El área de memoria denominada time retiene el valor que hay que almacenar en los registros del reloj, con lo que se especifica el tiempo que media entre dos notas emitidas. La subrutina sound hace algo semejante a la instrucción SOUND, por medio de OSWORD con A=7; soundtable es el bloque de parámetros para esta llamada OSWORD. Los datos para notetable y soundtable son establecidos por las sentencias en BASIC de las líneas 840 y 850.

Pulsando BREAK se detendrá este programa, restableciendo el contenido de EVENTV. Sin embargo, se puede hacer que continúe llamando a initialise.

El último programa que presentamos ilustra el modo de usar eventos para mover un sencillo dibujo de gráficos, concretamente un rectángulo, a través de la pantalla, una posición por cada evento. La frecuencia de movimiento depende, por ello, del valor asignado al reloj de eventos. Este valor se establece en la línea 1260. La definición del dibujo móvil, en forma de números PLOT, se almacena en shapetable y se establece por una sentencia DATA, mientras la línea 1220 coloca (POKE) los valores apropiados. Las líneas de la 1230 a la 1250 establecen EVENTV para apuntar a nuestro programa y activar el evento adecuado. Examinemos ahora el lenguaje máquina involucrado.

Está almacenado en la memoria en la dirección especificada por code%. Las líneas de la 200 a la 230 realizan la habitual operación de almacenamiento del registro, y las líneas de la 250 a la 280 hacen el trabajo por medio de una serie de llamadas a subrutinas. Finalmente, restauramos los registros y ejecutamos una RTS. Las líneas de la 370 a la 440 guardan el actual estado del cursor de gráficos y el actual GCOL usado, de modo que pueda ser almacenado después de tratado el evento.

De la 460 a la 520 se ocupan del movimiento del



Movimiento de gráficos

```
10 REM **** GRAFICOS CON USO RELOJ EVENTOS ****
30 MODE 1
40 PROCassemble
50 CALL clock
60 END
70 DEFPROCassemble
80 xpos=&70:ypos=&72
90 ?xpos=00:?(xpos+1)=0
100 ?ypos=100:?(ypos+1)=0
110 oswrch=&FFEE
120 osbyte=&FFF4
130 osword=&FFF1
140 DIM code% 600, time% 10
150
160 FOR pass=0 TO 2 STEP 2
170 P%=code%
180
190 [OPT pass
200 PHP
210 PHA
220 TXA:PHA
230 TYA:PHA
240
250 JSR preserve
260 JSR draw
270 JSR restore
280 JSR clock
290
300
310 PLA:TAY
320 PLA:TAX
330 PLA
340 PLP
350 RTS
360
370 .preserve
380 LDX #pgpos MOD 256
390 LDY #pgpos DIV 256
400 LDA #&0D
410 JSR osword
420 LDA &35B:STA ccol
430 LDA &35C:STA ccol+1
440 RTS
450
460 .draw
470 JSR gcol
480 JSR move
490 JSR shape
500 JSR incx
510 JSR incy
520 RTS
530
540 .restore
550 LDA ccol:STA &35B
560 LDA ccol+1:STA &35C
570 LDA #25:JSR oswrch
580 LDA #4:JSR oswrch
590 LDA cgpos:JSR oswrch
600 LDA cgpos+1:JSR oswrch
610 LDA cgpos+2:JSR oswrch
620 LDA cgpos+3:JSR oswrch
630 RTS
640
650 .move
660 LDA #25:JSR oswrch
670 LDA #4:JSR oswrch
680 LDA xpos:JSR oswrch
690 LDA xpos+1:JSR oswrch
700 LDA xpos+2:JSR oswrch
710 LDA xpos+3:JSR oswrch
720 RTS
730
740 .shape
750 LDX#24:LDY #0
760 .sloop LDA shapetable,Y
770 JSR oswrch
780 INY:DEX
790 BNE sloop
800 RTS
810
820 .incx CLC
830 LDA #2:ADC &70:STA &70
840 LDA #0:ADC &71:STA &71
850 RTS
860 .incy CLC
870 LDA #2:ADC &72:STA &72
880 LDA #0:ADC &73:STA &73
890 LDA &73:CMP #3:BCS overflow
900 RTS
910 .overflow
920 LDA #0:STA &72:STA &73
930 LDA #0:STA &70:STA &71
940 RTS
950
960 .clock
970 LDX #time% MOD 256
980 LDY #time% DIV 256
990 LDA #4
1000 JSR osword
1010 RTS
1020
1030
1040 .gcol
1050 LDA #18:JSR oswrch
1060 LDA #3:JSR oswrch
1070 LDA #1:JSR oswrch
1080 RTS
1090
1100
1110 .ccol EQUD &0000
1120 .pgpos EQUD &00000000
1130 .cgpos EQUD &00000000
1140 .shapetable
1150 EQUD &00000000
1160 EQUD &00000000
1170 EQUD &00000000
1180 EQUD &00000000
1190 EQUD &00000000
1200 EQUD &00000000
1210 ]:NEXT
1220 FOR I%=0 TO 23:READ data:?(shapetable+I%)=data:NEXT
1230 ?&220=code% MOD 256
1240 ?&221=code% DIV 256
1250 *FX14,5
1260 !time%=&FFFFFFEF:time%?4=&FF
1270 ENDPROC
1280 DATA 25,1,60,0,0,0,25,1,0,0,60,0,25,1,&C4,255,0,0,
     25,1,0,0,&C4,255
```

dibujo. De la 540 a la 630 restablecen los colores de los gráficos y la posición del cursor de gráficos al estado que tenían cuando entramos en la rutina.

De la 650 a la 720 emplean la rutina OSWRCH para ejecutar la instrucción equivalente a MOVE del BASIC según las posiciones *x* e *y* deseadas. Estas posiciones se almacenan en xpos y xpos+1 para la abscisa *x* e ypos e ypos+1 para la ordenada *y*. Las líneas de la 740 a la 800 dibujan la figura, de nuevo a través de OSWRCH. El registro Y se emplea como registro índice para acceder a la tabla de bytes que representa el dibujo (shapetable). Los bytes se envían como una corriente a través de OSWRCH.

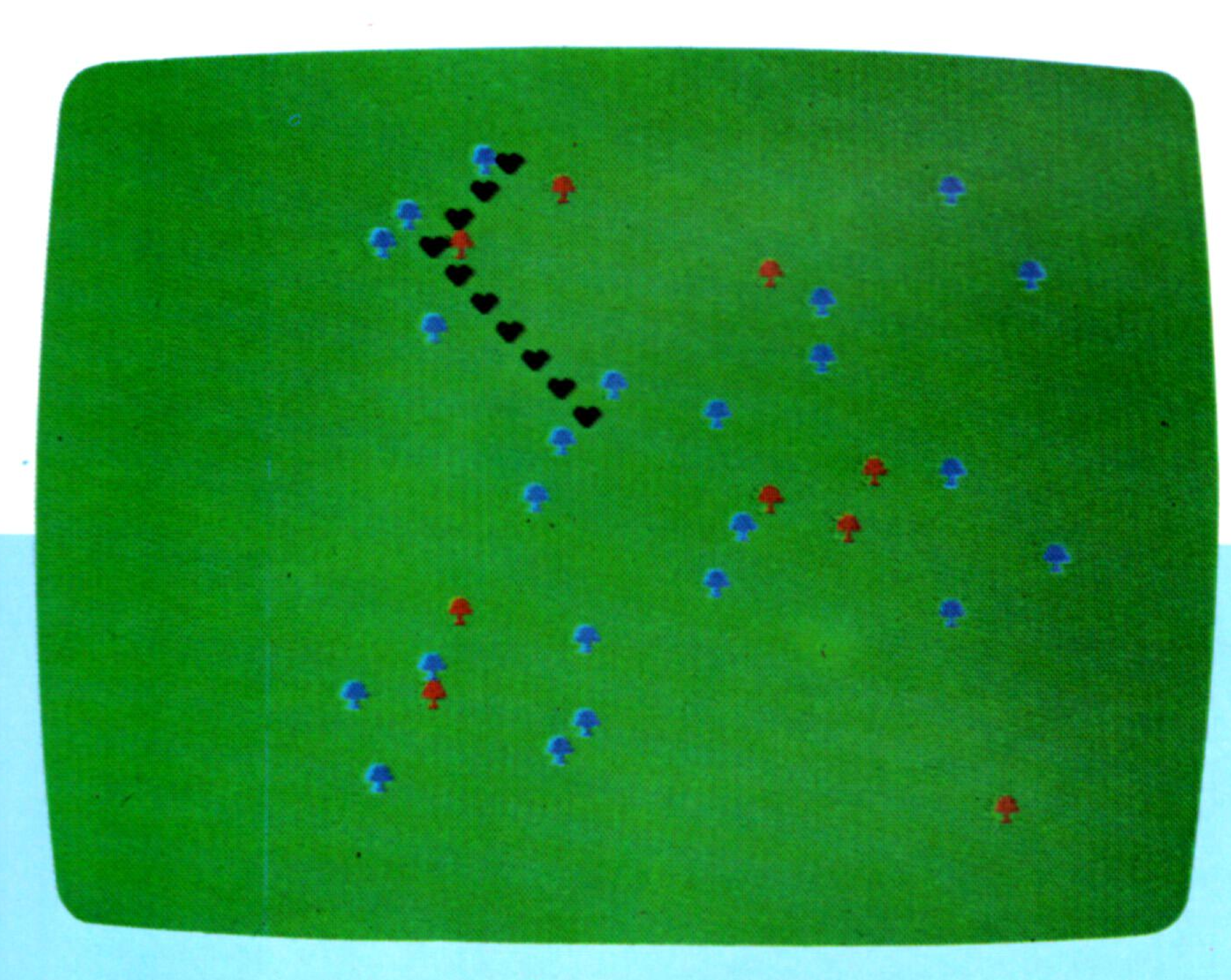
Dado que el dibujo se mueve por la pantalla, es claro que deben existir rutinas para actualizar las posiciones *x* e *y* cada vez que sucede el evento. De esto se encargan las rutinas incx e incy en las líneas de la 820 a la 900. Una vez que la caja alcanza la parte superior de la pantalla, la rutina overflow restablece las coordenadas *x* e *y* a cero. La rutina clock emplea OSWORD con A=4 para accionar el reloj de eventos, y la rutina gcol hace algo equivalente a GCOL3,1 del BASIC.

Hay varios problemas asociados a este programa. Ejecute el código máquina, y siempre y cuando la pantalla no se desplace (*scroll*), el dibujo se moverá bastante bien por la pantalla. Se pueden realizar otras operaciones mientras esto sucede, y todo es correcto. Pero si se acelera el movimiento incrementando el valor del área de memoria time, y aunque el dibujo se mueve cuando se lista parte del programa, puede que aparezcan caracteres extraños en la pantalla. La razón de todo esto es sencilla: el conflicto entre nuestra rutina que es entrada cuando sucede el evento, y el comportamiento habitual del OS cuando se hace el listado. Ambos requieren el empleo de OSWRCH y lo que sucede es que cuando la rutina de listado del OS es interrumpida por nuestro evento, OSWRCH se deja en un estado que no espera la rutina de listado.

Si se interrumpe OSWRCH y es llamada por la rutina de interrupción, entonces no puede obtenerse una salida de la llamada OSWRCH de la rutina de interrupción. Se dice que OSWRCH es *no-reentrante*. Por esto Acorn sugiere que no es recomendable el uso de rutinas OS en las rutinas de tratamiento de eventos y de interrupciones. Pero usted *puede* usarlas, siempre y cuando lo haga con cuidado. Estos problemas pueden ser superados mediante la renuncia al uso de cualquier rutina OS cuando se trate de una rutina de tratamiento de eventos o interrupción. Pero existen alternativas tales como entrar datos directamente en el área RAM de la pantalla para dibujar gráficos, y ésta es una de las pocas veces en que las llamadas de una rutina que evita la ROM es una buena idea.

Serpiente

Esta versión del conocido juego escrita en BASIC para el Thomson MO5 debe grabarse tal cual en cassette, utilizando la instrucción SAVE"<SERPIENTE>"



En este juego, usted es una serpiente que se desliza ondulando por la pantalla. El cambio de dirección se consigue pulsando cualquier tecla. Para poder desplazarse es preciso que se alimente. Felizmente se halla rodeado por gran cantidad de setas. Pero ¡cuidado! Si bien las setas azules son excelentes, ha de evitar las setas rojas, puesto que son venenosas. Cada seta azul le proporciona las calorías necesarias para avanzar diez líneas. ¡Procure no morirse de hambre pero sin acabar envenenado!

```
10 REM *****************
20 REM *   SERPIENTE   *
30 REM *****************
40 CLEAR ,,2
50 GOSUB 540
60 D$=INKEY$
70 IF D$<>"" THEN D=-D
80 X=X+D
90 IF X<0 THEN X=1
100 IF X>39 THEN X=38
110 IF POINT (X*8+4,Y*8+4)=1 THEN 260
120 IF POINT (X*8+4,Y*8+4)=4 THEN S=S+10:
    H=H+10:BEEP
130 LOCATE X,Y
140 COLOR 0
150 PRINT S$;
160 LOCATE 0,24,0
170 PRINT
180 LOCATE INT(RND*38)+1,24
190 COLOR 4
200 PRINT C$;
210 IF RND>0.5 THEN LOCATE INT (RND*38)+1
    ,24:COLOR 1:PRINT C$;
220 S=S-1
230 IF S=0 THEN 260
240 H=H+1
250 GOTO 60
260 BEEP
270 LOCATE 0,24
280 PRINT
290 LOCATE X,Y-1
300 COLOR 5
310 PRINT S$;
320 FOR I=1 TO 5
330 BEEP
340 FOR J=1 TO 50
350 NEXT J
360 NEXT I
370 IF INKEY$<>"" THEN 370
380 IF H>R THEN R=H
390 LOCATE 4,20
400 COLOR 0
410 PRINT "PUNTOS :";H;
420 LOCATE 23,20
430 PRINT "RECORD :";R;
440 LOCATE 15,23
450 PRINT "OTRA ?"
460 D$=INKEY$
470 IF D$="" THEN 460
480 IF D$<>"N" THEN 520
490 SCREEN 4,6,6
500 CLS
510 END
520 GOSUB 610
530 GOTO 60
540 CLS
550 SCREEN 2,2,2
560 DEFINT A-Z
570 DEFGR$(0)=0,0,102,255,219,126,60,24
580 DEFGR$(1)=60,126,126,255,255,24,24,
    60
590 S$=GR$(0)
600 C$=GR$(1)
610 CLS
620 S=100
630 H=0
640 D=1
650 X=19
660 Y=10
670 RETURN
```

Editorial Delta, S.A.



10080
9 788485 822836